Aspects of Guaranteed Error Control in Computations for Partial Differential Equations

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Dr. rer. nat.
im Fach Mathematik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät II
Humboldt-Universität zu Berlin

von
Dipl.-Math. Christian Merdon

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:
Prof. Dr. Elmar Kulke

Gutachter:
1. Prof. Dr. Carsten Carstensen, Humboldt-Universität zu Berlin
2. Prof. Dr. Andreas Schröder, Universität Salzburg
3. Prof. Dr. Stefan Funken, Universität Ulm

Tag der mündlichen Prüfung: 02. August 2013

Abstract

This thesis studies guaranteed error control for elliptic partial differential equations on the basis of the Poisson model problem, the Stokes equations and the obstacle problem. The error control derives guaranteed upper bounds for the energy error between the exact solution and different finite element discretisations, namely conforming and nonconforming first-order approximations.

The unified approach expresses the energy error by dual norms of one or more residuals plus computable extra terms, such as oscillations of the given data, with explicit constants. There exist various techniques for the estimation of the dual norms of such residuals. This thesis focuses on equilibration error estimators based on Raviart-Thomas finite elements, which permit efficient guaranteed upper bounds. The proposed postprocessing in this thesis considerably increases their efficiency at almost no additional computational costs. Nonconforming finite element methods also give rise to a nonconsistency residual that permits alternative treatment by conforming interpolations.

A side aspect concerns the explicit residual-based error estimator that usually yields cheap and optimal refinement indicators for adaptive mesh refinement but not very sharp guaranteed upper bounds. A novel variant of the residual-based error estimator, based on the Luce-Wohlmuth equilibration design, leads to highly improved reliability constants.

A large number of numerical experiments compares all implemented error estimators and provides evidence that efficient and guaranteed error control in the energy norm is indeed possible in all model problems under consideration. Particularly, one model problem demonstrates how to extend the error estimators for guaranteed error control on domains with curved boundary.

Zusammenfassung

Diese Arbeit behandelt garantierte Fehlerkontrolle für elliptische partielle Differentialgleichungen anhand des Poisson-Modellproblems, des Stokes-Problems und des Hindernisproblems. Hierzu werden garantierte obere Schranken für den Energiefehler zwischen exakter Lösung und diskreten Finite-Elemente-Approximationen erster Ordnung entwickelt.

Ein verallgemeinerter Ansatz drückt den Energiefehler durch Dualnormen eines oder mehrerer Residuen aus. Hinzu kommen berechenbare Zusatzterme, wie Oszillationen der gegebenen Daten, mit expliziten Konstanten. Für die Abschätzung der Dualnormen der Residuen existieren viele verschiedene Techniken. Diese Arbeit beschäftigt sich vorrangig mit Equilibrierungsschätzern, basierend auf Raviart-Thomas-Elementen, welche effiziente garantierte obere Schranken ermöglichen. Diese Schätzer werden mit einem Postprocessing-Verfahren kombiniert, das deren Effizienz mit geringem zusätzlichen Rechenaufwand deutlich verbessert. Nichtkonforme Finite-Elemente-Methoden erzeugen zusätzlich ein Inkonsistenzresiduum, dessen Dualnorm mit Hilfe diverser konformer Approximationen abgeschätzt wird.

Ein Nebenaspekt der Arbeit betrifft den expliziten residuen-basierten Fehlerschätzer, der für gewöhnlich optimale und leicht zu berechnende Verfeinerungsindikatoren für das adaptive Netzdesign liefert, aber nur schlechte garantierte obere Schranken. Eine neue Variante, die auf den equilibrierten Flüssen des Luce-Wohlmuth-Fehlerschätzers basiert, führt zu stark verbesserten Zuverlässigkeitskonstanten.

Eine Vielzahl numerischer Experimente vergleicht alle implementierten Fehlerschätzer und zeigt, dass effiziente und garantierte Fehlerkontrolle in allen vorliegenden Modellproblemen möglich ist. Insbesondere zeigt ein Modellproblem, wie die Fehlerschätzer erweitert werden können, um auch auf Gebieten mit gekrümmten Rändern garantierte obere Schranken zu liefern.

Contents

1 Introduction

This thesis studies finite element methods for elliptic partial differential equations of second order and derives sharp guaranteed upper bounds for the energy error between the exact and discrete solution for three exemplary problem classes. The introduction proceeds with the explanation and motivation of the mentioned keywords and an overview of the content of this thesis. The last part of the introduction draws some conclusions and gives an outlook for possible further applications.

Partial Differential Equations of Second Order

Partial differential equations allow mathematical modeling of various physical processes. The Poisson model problem is the most fundamental elliptic partial differential equation and arises in numerous applications in the field of potential theory. The strong formulation of the Poisson model problem in 2D, for given data $f : \Omega \to \mathbb{R}$ and homogeneous Dirichlet boundary data along $\partial\Omega$, seeks $u \in C^2(\Omega)$ with

$$-\operatorname{div}(\nabla u) = f \text{ in } \Omega \quad \text{and} \quad u = 0 \text{ on } \partial\Omega. \tag{1.1}$$

The function u represents an electric field potential in static electromagnetism, or a hydraulic head in steady-state groundwater flow, or a gravitational potential in classical mechanics. The right-hand side f in these applications determines (up to constants) the charge density in static electromagnetism, or the mass distribution in gravitation. For steady-state groundwater flow, the hydraulic head is a harmonic function, hence $f \equiv 0$ but with inhomogeneous boundary conditions instead. Moreover, every linear second order elliptic partial differential equation with constant coefficients can be transformed into a Poisson problem. Therefore, it is reasonable to study this model problem thoroughly. Real world applications replace the zero boundary conditions in (1.1) by Dirichlet boundary conditions, which fix certain values of u along the boundary of Ω, or Neumann boundary conditions, which prescribe the normal component of ∇u. Additional time-dependent derivatives lead to parabolic (e.g. the heat equation) or hyperbolic (e.g. the wave equation) partial differential equations of second order.

Finite Element Methods

Finite element methods (FEMs) are very popular and flexible tools for the numerical approximation of solutions of partial differential equations in computational mechanics. These methods approximate the solution u or its stress tensor $\sigma = \nabla u$ by piecewise polynomials on a regular triangulation of the domain Ω into triangles (in 2D) or tetrahedra (in 3D). To assess the quality of the discrete solution, the a posteriori error estimation is an important field of interest. It enables adaptive mesh design and stopping criteria for the discretisation process. Its importance has attracted high attention over the last decades

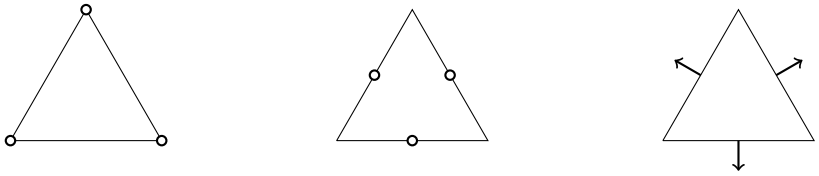

Figure 1.1: Schematic visualisation of the Lagrange (left), Crouzeix-Raviart (middle) and Raviart-Thomas (right) finite element.

and led to textbooks such as Verfürth (1996); Ainsworth and Oden (2000); Han (2005); Repin (2008) and to special chapters in standard textbooks on FEM such as Babuška and Strouboulis (2001); Braess (2007); Brenner and Scott (2008).

The characteristic of a finite element method is the choice of polynomials and their continuity or regularity properties at certain nodes (degrees of freedom) of the triangulation. This work studies mainly the $\mathcal{P}_1$ conforming FEM, the Crouzeix-Raviart nonconforming FEM and the Raviart-Thomas mixed FEM of lowest order. Figure 1.1 displays a schematic view on their local degrees of freedom on a single triangle.

Each method leads to a discrete stress tensor σ_h, which is an approximation of the exact stress tensor σ. The overall goal is to design error estimators for the energy norm difference $\|\sigma - \sigma_h\|_{L^2(\Omega)} \coloneqq \int_\Omega |\sigma - \sigma_h|^2 \,\mathrm{d}x$. Reliability and efficiency of an error estimator η means existence of equivalence constants c_1 and c_2 independent of the mesh size and up to higher-order terms (hot), such that

$$\text{hot} + c_1\eta \leqslant \|\sigma - \sigma_h\|_{L^2(\Omega)} \leqslant c_2\eta + \text{hot}. \tag{1.2}$$

Adaptive refinement based on local contributions to η leads to improved convergence rates of the energy error and so to more economic use of CPU time and the memory that is needed to represent a solution with a certain quality. Figure 1.2 compares the energy error $|||u - u_h||| \coloneqq \|\sigma - \sigma_h\|_{L^2(\Omega)}$ between the exact flux and the discrete flux of the $\mathcal{P}_1$ conforming finite element method for some Poisson model problem on an L-shaped domain. The adaptive mesh refinement automatically detects singularities like the reentrant corner in this example and leads to optimal convergence rates. However, the proper choice of refinement indicators, at least for the Poisson model problem, is well-understood, so this is only a side aspect of this thesis.

Guaranteed Upper Error Bounds

The main focus in this thesis lies on guaranteed upper bounds (called "error majorants" by Repin (1999)), i.e., $c_2 = 1$ and computable terms of higher order in (1.2). In all model problems under consideration, the error analysis leads to residuals of the form

$$\mathrm{Res}(v) \coloneqq \int_\Omega fv \,\mathrm{d}x + \int_\Omega \sigma_h \cdot \nabla v \,\mathrm{d}x \quad \text{for } v \in V \coloneqq H^1_0(\Omega) \tag{1.3}$$

and associated dual norms

$$|||\mathrm{Res}|||_{Z^\star} \coloneqq \sup_{v \in Z} \mathrm{Res}(v) / |||v|||$$

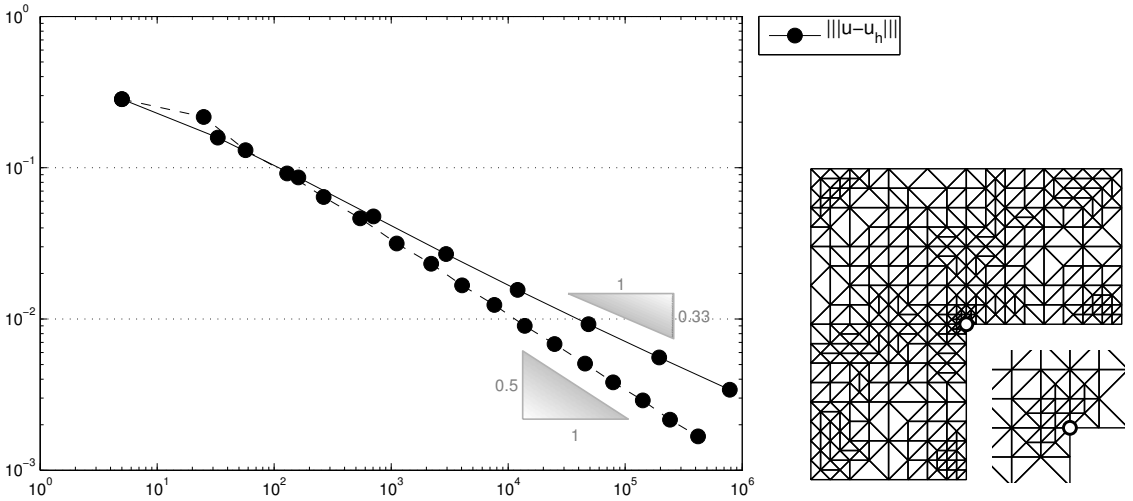

Figure 1.2: Convergence history for the energy error $|||u - u_h|||$ for the $\mathcal{P}_1$-conforming finite element solution u_h of the Poisson model problem (1.1) with $f \equiv 1$ on uniform (solid line) and adaptive (dashed line) meshes as a function of the number of degrees of freedom. The right image shows the adaptive mesh on level 4.

with respect to some test function space $Z \subseteq H^1$ and an energy norm $|||\cdot|||$. For instance, the Poisson model problem (1.1) and its discrete $\mathcal{P}_1$ conforming finite element solution u_h with discrete flux $\sigma_h := \nabla u_h$ lead to $Z = V$, $|||\cdot||| := \|\nabla\cdot\|_{L^2(\Omega)}$ and $\|\sigma - \sigma_h\|_{L^2(\Omega)} = |||u - u_h||| = |||\mathrm{Res}|||_{V^\star}$. In other applications or for inhomogeneous Dirichlet data further overhead terms appear. Although they are of higher order for smooth data, this thesis includes a thorough analysis and derives explicit constants for these additional terms to allow for guaranteed error control also on arbitrary coarse meshes and for nonsmooth data.

Equilibration Error Estimators

The most efficient guaranteed upper bounds of $|||\mathrm{Res}|||_\star$ involve the design of some equilibrated quantity q that has an L^2-measurable divergence $\operatorname{div} q \in L^2(\Omega)$, i.e. $q \in H(\operatorname{div}, \Omega)$. The terminus "equilibrated" means that $f + \operatorname{div} q$ is small and spawns overhead terms of higher order that are computable oscillations of f and vanish if f is constant. For the Poisson model problem, this leads to

$$|||\mathrm{Res}|||_{V^\star} = |||\operatorname{div}(\sigma_h - q)|||_{V^\star} + \mathrm{hot} \leqslant \|\sigma_h - q\|_{L^2(\Omega)} + \mathrm{hot}. \tag{1.4}$$

Some popular designs for q were suggested by Ainsworth and Oden (2000), Braess (2007) or Luce and Wohlmuth (2004) and also the design in Carstensen and Funken (1999) can be identified as an equilibration error estimator. Minimisation of the right-hand side of (1.4) over some discrete subspace of $H(\operatorname{div}, \Omega)$ leads to mixed or least-square FEMs or the dual error majorants of Repin (1999). Equilibration error estimates are usually quite

sharp, however the estimate in (1.4) is suboptimal, since

$$\|\!|\!|\operatorname{div}(\sigma_h - q)|\!|\!|_\star = \sup_{\varphi\in V} \int_\Omega (\sigma_h - q)\cdot\nabla\varphi\,\mathrm{d}x = \min_{v\in H^1(\Omega)} \|\sigma_h - q - \operatorname{Curl} v\|_{L^2(\Omega)}\,. \tag{1.5}$$

In fact, this is the nature of the Helmholtz decomposition and coincides with the majorant theory of Repin (1999), since every $\widehat{q} := q - \operatorname{Curl} v$ is again an equilibrated quantity in (1.4) and an error majorant in the sense of Repin. However, the algorithmic exploitation of this identity is not so clear and the computation of the exact $q = \sigma$ or $\operatorname{Curl} v$ is as expensive as the solve of the original problem. This thesis suggests to compute an equilibrated quantity q after Braess or Luce-Wohlmuth first. Then a novel postprocessing computes and subtracts some discrete $\operatorname{Curl} v$ and so leads to increased efficiency at low costs.

Figure 1.3 displays efficiency indices $\|\sigma_h - q - \operatorname{Curl} v\|_{L^2(\Omega)} / \|\sigma - \sigma_h\|_{L^2(\Omega)}$ for some equilibration error estimators after Braess (B) or Luce-Wohlmuth (LW) and the post-processed quantities (Br(1), Brr(3), LW(1)). In this example, the efficiency indices are improved by a factor ten, namely from 1.3 to 1.03 in case of the Braess error estimator. Hence, Brr(3) is an almost exact guaranteed upper error bound. More numerical benchmark problems are studied in Chapter 4 also with mixed boundary conditions and discontinuous diffusion coefficients. Moreover, domains with curved boundaries, which cannot be approximated exactly with triangulations into triangles, are considered.

Nonconsistency Residual for Nonconforming Approximations

While the conforming solutions u_h are in $H^1_0(\Omega)$, the nonconforming approximations u_{CR} by Crouzeix-Raviart finite element methods are not. So the latter ones only allow for a piecewise gradient $\nabla_{\mathrm{NC}} u_{\mathrm{CR}}$, which has also a Curl contribution in its Helmholtz decomposition. In the error analysis for the 2D Poisson model problem (1.1), this leads to a second nonconsistency residual of the form

$$\operatorname{Res}_{\mathrm{NC}}(v) := -\int_\Omega \nabla_{\mathrm{NC}} u_{\mathrm{CR}}\cdot\operatorname{Curl} v\,\mathrm{d}x = \int_\Omega \operatorname{Curl}_{\mathrm{NC}} u_{\mathrm{CR}}\cdot\nabla v\,\mathrm{d}x \quad \text{for } v\in H^1(\Omega).$$

This residual has the form of (1.3) with $f\equiv 0$ and $\sigma_h = \operatorname{Curl} u_{\mathrm{CR}}$ (and hence permits the application of equilibration error estimators), but there is also the characterisation

$$\|\!|\!|\operatorname{Res}_{\mathrm{NC}}|\!|\!|_{H^1(\Omega)^\star} = \min_{v\in H^1_0(\Omega)} \|\nabla_{\mathrm{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)}\,.$$

This implies that any (discrete) conforming approximation $v\in H^1_0(\Omega)$ of u_{CR} leads to a guaranteed upper bound of $\|\!|\!|\operatorname{Res}_{\mathrm{NC}}|\!|\!|_{H^1(\Omega)^\star}$. Section 4.4 explains several possibilities, also for mixed boundary conditions. One of the most efficient designs in 2D employs the red-refinement and solves local one-dimensional problems to define some piecewise linear v with respect to the red-refined triangulation.

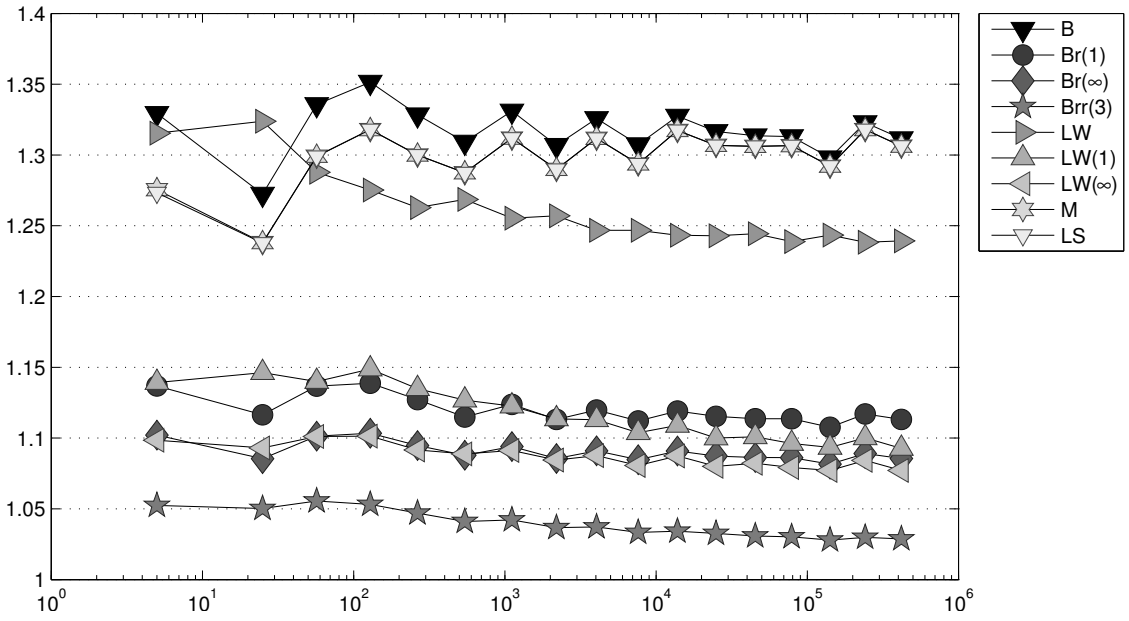

Figure 1.3: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes for the Poisson model problem (1.1) with $f \equiv 1$.

Stokes Problem

For given data $f : \Omega \to \mathbb{R}^2$, the 2D Stokes problem seeks a velocity field $u \in H^1(\Omega;\mathbb{R}^2)$ and a pressure $p \in L^2(\Omega)$, such that

$$-\operatorname{div}(\nabla u) - \nabla p = f \quad \text{and} \quad \operatorname{div} u = 0 \text{ in } \Omega \quad \text{while} \quad u = 0 \text{ on } \Gamma_D. \tag{1.6}$$

This problem describes the motion of Newtonian fluids, like water. The constraint $\operatorname{div} u = 0$ in (1.6) implies the incompressibility of the fluid, while f is a force that accelerates or decelerates the flow. Since only the gradient of p enters, it is unique up to some constant which is commonly fixed with the condition $\int_\Omega p \,\mathrm{d}x = 0$. The stress tensor $\sigma = \nabla u + p\mathbb{I}$ solves the Poisson problem $-\operatorname{div}\sigma = f$ and the error analysis involves a residual of the type (1.3), but, due to the side constraint, is tested with divergence-free functions Z. In the design of guaranteed upper bounds, this property allows to restrict the investigation to the deviatoric part, i.e.,

$$|||\operatorname{div}(\sigma_h - q)|||_{Z^\star} = \min_{v\in H^1(\Omega)} \|\operatorname{dev}(\sigma_h - q - \operatorname{Curl} v)\|_{L^2(\Omega)}\,.$$

The divergence constraint also leads to some difficulties in the discretisation. For existence and uniqueness of solutions, the discretisation spaces of $u_h \in X_h$ and $p_h \in Y_h$ have to satisfy an inf-sup-condition, which reads

$$0 < c_0 \coloneqq \inf_{q_h\in Y_h\setminus\{0\}} \sup_{v_h\in X_h} \frac{\int_\Omega q_h \operatorname{div}(v_h)}{\|\mathrm{D}v_h\|_{L^2(\Omega)}\,\|q_h\|_{L^2(\Omega)}}. \tag{1.7}$$

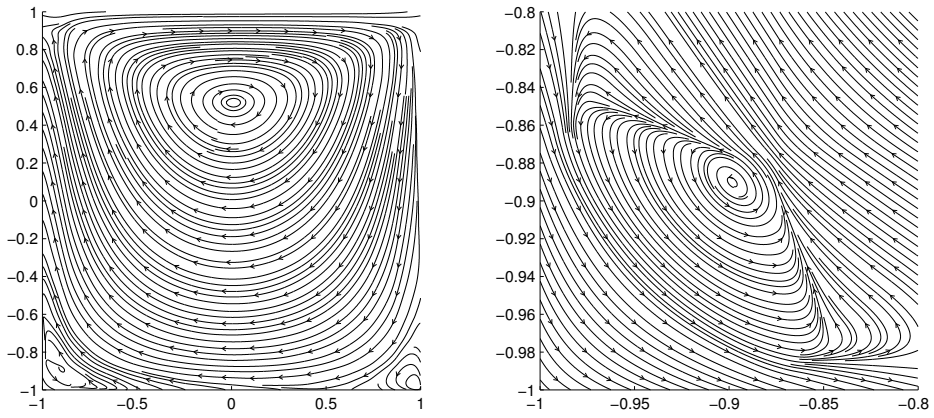

Figure 1.4: Streamlines of Stokes Example with driven cavity on a square domain and zoom-in for the Moffatt eddies in the lower left corner.

It turns out that $\mathcal{P}_1$ conforming elements for X_h and $\mathcal{P}_0$ or $\mathcal{P}_1$ conforming elements for Y_h yield $c_0 = 0$, and hence lead to an unstable discretisation scheme. The simplest stable finite element method is the mini finite element method that enriches the velocity space with cubic element bubble functions. Another popular finite element for this application is the Taylor-Hood finite element that pairs $\mathcal{P}_2$ elements (or the isomorphic $\mathcal{P}_1$ elements on the red-refinement) for the velocity with $\mathcal{P}_1$ elements for the pressure.

The nonconforming discretisation, however, is stable also for the lowest-order Crouzeix-Raviart finite elements. Moreover, the selection of divergence-free Crouzeix-Raviart functions also allows the elimination of the pressure and the $\operatorname{div} u = 0$ constraint. Figure 1.4 shows the velocity field and streamlines for some simulation of a lid-driven cavity flow with $f \equiv 0$ and inhomogeneous Dirichlet data. The zoom of the lower right corner reveals the characteristic Moffatt eddies.

Variational Inequalities

Variational inequalities involve an energy functional that is to be minimised amongst a certain (convex) set of admissible functions. The obstacle problem in Chapter 6 is the prototype of such a variational inequality and minimises the energy

$$E(v) \coloneqq \frac{1}{2}\int_\Omega |\nabla v|^2 \,\mathrm{d}x - \int_\Omega fv \,\mathrm{d}x \tag{1.8}$$

amongst the set of functions

$$\mathcal{K} \coloneqq \left\{ v \in H^1_0(\Omega) \,|\, \chi \leqslant v \right\} \text{ for some obstacle function } \chi.$$

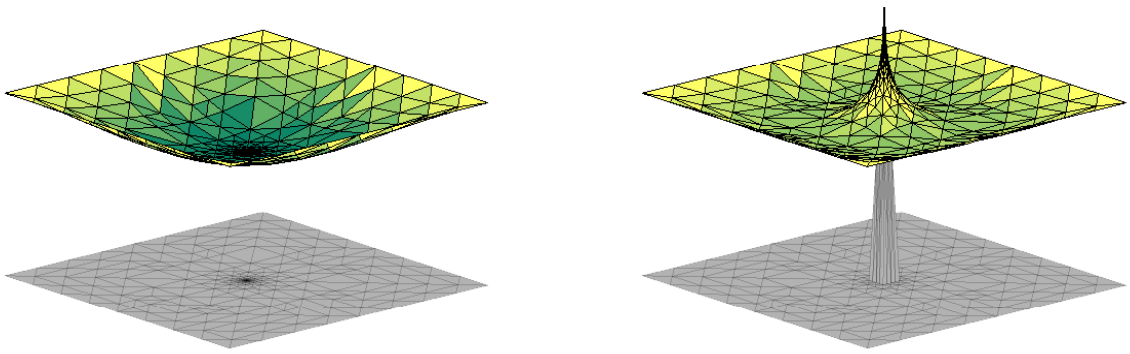

Figure 1.5: Energy minimiser u_h without an obstacle constraint (left) and for some (gray-shaded) cusp-shaped obstacle $\chi \leqslant u_h$ (right) and $f \equiv -2$.

The weak formulation characterises the exact solution $u \in \mathcal{K}$ by the variational inequality

$$\int_\Omega \nabla u \cdot \nabla(u - v)\,\mathrm{d}x \leqslant \int_\Omega f(u - v)\,\mathrm{d}x \quad \text{for all } v \in \mathcal{K}.$$

Details can be found in the literature, e.g. in Kinderlehrer and Stampacchia (1980). Figure 1.5 displays two solutions for the energy minimisation problem with $f \equiv -2$. The right picture shows the minimiser for some cusp-shaped obstacle and the left picture shows the minimiser without an obstacle constraint which is in fact the solution of the Poisson problem with right-hand side f.

While the unconstrained solutions u and u_h satisfy $\|\!|\mathrm{Res}|\!\|_\star = \|\!|u - u_h|\!\|$, this identity does not hold for the constrained solutions of the obstacle problem. The residual of (1.3) does not take into account the contact force in the part of the domain where u_h touches the obstacle χ. The revelating view of Braess (2005) on the obstacle problem identifies the discrete solution u_h as the solution of an auxiliary Poisson problem for the right-hand side $f - \Lambda_h$ with some nonunique discrete contact force Λ_h. Then, the energy error equals

$$\|\!|u - u_h|\!\| \leqslant \|\!|\mathrm{Res}_{\mathrm{aux}}|\!\|_\star + \text{overhead terms},$$

where $\mathrm{Res}_{\mathrm{aux}}$ is the residual for the auxiliary Poisson problem. The discrete contact force Λ_h, however, has to satisfy certain properties, and there are few suggestions in the literature how to realise them. The choice of Carstensen and Merdon (2012) is improved in this thesis in some aspects, e.g., rigorous analysis and explicit constants for all overhead terms and the considerations of extra errors due to inexact solve. This allows guaranteed error control also for inexact discrete solutions (an exact solve is in general infeasible here) that violate the discrete complementary conditions. Efficiency is proven for affine obstacles χ.

Structure of this Thesis

This thesis is structured as follows. Chapter 2 explains all theoretical preliminaries such as Sobolev spaces, finite element spaces, triangulations and the three finite element methods

of Figure 1.1 for the Poisson model problem. It also gives a brief introduction to the data structures and the MATLAB implementation.

Chapter 3 motivates and studies residuals of the form (1.3). The chapter continues with an explanation of the equilibration error estimator design after Braess and Luce-Wohlmuth and of the Curl postprocessing. The final part of the chapter concerns the standard residual-based error estimator and its efficiency as well as a novel variant with explicit reliability constants that render it a guaranteed upper bound.

Chapter 4 studies a generalised Poisson model problem with mixed homogeneous boundary conditions and a discontinues diffusion tensor, also for three dimensions and domains with curved boundaries. For all three finite element methods under considerations, guaranteed upper bounds for the energy error are derived and validated in many numerical benchmark examples from the literature.

Chapter 5 concerns the Stokes problem and commences with the discretisation by the mini finite element method and its error analysis, which is valid also for other conforming discretisations. The second part studies the nonconforming discretisation with Crouzeix-Raviart elements. The efficiency of the derived guaranteed upper bounds for both methods is examined in five benchmark problems.

Chapter 6 studies the obstacle problem. The first part introduces the setting of this variational inequality and its conforming discretisation. Then, the details of the Braess methodology are explained to achieve guaranteed upper bounds for the energy error with several overhead terms and explicit constants. Moreover, efficiency for affine obstacles is proven and verified in some numerical benchmark examples.

Appendix A shows the main parts of the MATLAB implementation and gives an overview of the content of the data carrier that comes with this thesis.

Appendix B lists and explains all notation that is used throughout the thesis.

Conclusions and Outlook

The main conclusion of this thesis is that guaranteed error control with sharp efficiency indices is indeed possible for all problems under considerations with a typical overestimation of the guaranteed error bound by a factor 1 to 4.

The suggested postprocessing allows improvement of the efficiency of any equilibration error estimator at very low extra effort. Naturally, the gain of efficiency is limited by the influence of the overhead terms. They are the reason for the slightly worse efficiency indices in the numerical examples for the Stokes problem (about 1 to 4) compared to the efficiency indices for Poisson problems (close to 1). The rather pessimistic values for the constant c_0 from (1.7), which enter the guaranteed upper bounds, lead to some unnecessary overestimation. Better knowledge and improved guaranteed lower bounds for c_0, e.g., by numerical solve of the associated general eigenvalue problem, might improve the efficiency of the error estimators dramatically. For obstacle problems, at least those with affine obstacles, the overhead terms are of higher order and yield efficiency indices between 1 and 2. The combination of the postprocessing with the mean correction is able to reduce the influence of the oscillation terms on coarse meshes.

However, the theory of this thesis is not limited to these three problems and also not limited to error control in terms of energy norms. Further applications include linear elasticity, convection-diffusion-reaction partial differential equations or nonlinear

problems as Poisson problems with friction. All techniques also transfer to parabolic partial differential equations. Furthermore, goal-oriented error estimation is possible through duality techniques and Riesz representation of the goal functional. The efficiency of the error estimators for the energy norm directly influences the sharpness of the guaranteed bounds for the goal error.

A further challenge, vital to nonlinear problems, is the inexact solve. In this case the discrete solution looses its Galerkin orthogonality property which is a prerequisite for the equilibration designs of Braess and Luce-Wohlmuth. However, as in the case of obstacle problems, the discrete solution solves a perturbed problem with a different right-hand side that can be analysed instead and causes another extra term in the guaranteed upper bound that measures the truncation error of the iterative solver.

Acknowledgements

These last lines of the introduction I want to dedicate to all the people that helped me to realise this thesis. That includes foremost my doctoral advisor Prof. Carsten Carstensen who taught me almost everything I know now about finite element methods, error estimation and paper writing. I also want to thank my colleague Wolfgang Boiger who became a valuable friend during our research stays in the Republic of Korea in 2009 and 2010. I learned so much practical things from him, like git and tikz, that helped me to write up this thesis fast and efficiently. Finally, I want to thank all my other colleagues and friends and my family for their support during the last years.

2 Theoretical Preliminaries

This chapter recalls theoretical preliminaries needed for the mathematical modeling and the numerical analysis in the thesis.

2.1 Functional Analysis for Sobolev Spaces

The following section gives a very short introduction to Sobolev spaces. A complete and more detailed introduction can be found in textbooks like Grisvard (1985) or Evans and Gariepy (1992).

2.1.1 Sobolev Spaces

Here and throughout, $\Omega \subset \mathbb{R}^n$ denotes some Lipschitz domain in $n = 2$ or $n = 3$ dimensions with polygonal or polyhedral boundary $\partial\Omega$ and outer normal vector ν. The boundary may consist of some closed Dirichlet part Γ_D with positive surface measure and some (possibly empty) Neumann part $\Gamma_N := \partial\Omega \backslash \Gamma_D$. The space $L^p(\Omega;\mathbb{R}^m)$ denotes the Lebesgue spaces of L^p-integrable functions over Ω with m components and $L^p_{\mathrm{loc}}(\Omega)$ contains all functions that are L^p integrable on every open subset $\omega \subset\subset \Omega$ that is compactly contained in Ω. The function space $C^k_C(\Omega;\mathbb{R}^m)$ denotes all k times differentiable functions with compact support in Ω, while $C_D(\Omega;\mathbb{R}^m)$ denotes the continuous functions with zero boundary conditions along Γ_D.

Definition 2.1.1 (Weak Derivative)**.** *The function $g_j \in L^1_{\mathrm{loc}}(\Omega;\mathbb{R}^m)$ is called a* weak derivative *of $v \in L^1_{\mathrm{loc}}(\Omega;\mathbb{R}^m)$ with respect to x_j, $j = 1,\dots,n$, if the integration by parts formula*

$$\int_\Omega v \cdot \partial\varphi/\partial x_j \,\mathrm{d}x = -\int_\Omega g_j \cdot \varphi \,\mathrm{d}x \quad \text{holds for all } \varphi \in C^1_C(\Omega;\mathbb{R}^m).$$

In this case, g_j is abbreviated with $\partial v/\partial x_j$ and if all partial derivatives exist, $\mathrm{D}v = (\partial v/\partial x_1, \dots, \partial v/\partial x_n) \in L^1_{\mathrm{loc}}(\Omega;\mathbb{R}^{m\times n})$ denotes the weak gradient of v.

Definition 2.1.2 (Sobolev Spaces)**.** *The* Sobolev space H^1 *consists of all L^2 functions whose weak gradient is also in L^2, i.e.,*

$$H^1(\Omega;\mathbb{R}^m) := \left\{ v \in L^2(\Omega;\mathbb{R}^m) \,\middle|\, \mathrm{D}v \in L^2(\Omega;\mathbb{R}^{m\times n}) \right\}.$$

The space of all L^2 functions with divergence in L^2 reads

$$H(\mathrm{div},\Omega;\mathbb{R}^{m\times n}) := \left\{ v \in L^2(\Omega;\mathbb{R}^{m\times n}) \,\middle|\, \mathrm{div}\, v \in L^2(\Omega;\mathbb{R}^m) \right\}.$$

Definition 2.1.3 (Dual Space and Dual Norm)**.** *The* dual space $B^\star$ *of some Banach space* B *with respect to the norm* $\|\cdot\|$ *consists of all bounded linear functionals* $F : B \to \mathbb{R}$*, i.e.* $B^\star := L(B, \mathbb{R})$*, and the* dual norm *of some* $F \in B^\star$ *reads*

$$\|F\|_\star := \sup_{v \in B\setminus\{0\}} F(v)/\|v\| .$$

Remark 2.1.4. *The dual space of* $L^p(\Omega;\mathbb{R}^m)$ *is (by Hölder inequality) the space* $L^q(\Omega;\mathbb{R}^m)$ *with* $1/p + 1/q = 1$*.*

In the following, the codomain of a function space is omitted if $m = 1$, e.g., $L^2(\Omega;\mathbb{R}^1) = L^2(\Omega)$ and $H(\mathrm{div},\Omega;\mathbb{R}^{1\times n}) = H(\mathrm{div},\Omega)$. Moreover, $\nabla v = \mathrm{D}v$ denotes the derivative for scalar-valued functions $v \in H^1(\Omega)$.

2.1.2 Traces of Sobolev Functions

Since Lebesgue function are defined up to sets of measure zero, the existence of well-defined traces in $L^2(\partial\Omega)$ along the boundary $\partial\Omega$ is not obvious. However, under certain regularity assumptions they exist.

Theorem 2.1.5 (Existence of Traces (Evans and Gariepy, 1992; Temam, 2001))**.** *For a bounded Lipschitz domain* Ω *there exists a bounded linear operator* $T : H^1(\Omega) \to L^2(\partial\Omega)$*, called* trace operator*, such that*

(a) $T(v) = v$ *on* $\partial\Omega$ *for all* $v \in H^1(\Omega) \cap C(\overline{\Omega})$*, and,*
(b) for all $q \in H(\mathrm{div},\Omega)$ *and* $v \in H^1(\Omega)$*, it holds*

$$\int_{\partial\Omega} T(q\cdot\nu)v \,\mathrm{d}s = \int_\Omega v \,\mathrm{div}\, q \,\mathrm{d}x + \int_\Omega \mathrm{D}v \cdot q \,\mathrm{d}x.$$

Proof. The proof of (a) studies

$$T(v)(y) := \lim_{r\to 0} \fint_{B(r,y)\cap\Omega} v(x)\,\mathrm{d}x \quad \text{for } y \in \partial\Omega \text{ and } v \in H^1(\Omega)$$

and can be found in Evans and Gariepy (1992, Theorem 1 on page 133). A proof of (b) is given in Temam (2001, Theorem 1.2 on page 7) where $H(\mathrm{div},\Omega)$ is named as $E(\Omega)$. □

Remark 2.1.6. *Actually, the normal flux* $q\nu$ *in the left-hand side of (b) is an element of the dual space* $H^{-1/2}(\partial\Omega)$ *of* $H^{1/2}(\partial\Omega) \supset L^2(\partial\Omega)$*. Therefore, the integral is to be understood in a distributional sense. However, to facilitate a simple notation we abide by the integral notation and usually we consider* $H(\mathrm{div},\Omega)$ *functions with trace in* $L^2(\partial\Omega)$*. Moreover,* T *is omitted in the sequel.*

Definition 2.1.7. *The space of functions with zero trace along* Γ_D *reads*

$$H^1_D(\Omega) := \{v \in H^1(\Omega) \,\big|\, v = 0 \text{ along } \Gamma_D\}.$$

2.1.3 Basic Inequalities

This subsection collects some basic inequalities related to Sobolev spaces. Although these inequalities are formulated for scalar-valued functions here, they hold verbatimly for vector-valued functions.

Theorem 2.1.8 (Poincaré Inequality)**.** *Let Ω be a Lipschitz domain with C^1 boundary. There exists a constant $C_{\mathrm{P}}(\Omega)$ of Ω such that, for any function $v \in H^1(\Omega)$ with $\fint_\Omega v = 0$, it holds*

$$\|v\|_{L^2(\Omega)} \leqslant C_{\mathrm{P}}(\Omega) \operatorname{diam}(\Omega) \|\nabla v\|_{L^2(\Omega)}.$$

The constant $C_{\mathrm{P}}(\Omega)$ is invariant under rescaling of Ω.

Proof. A proof can be found in Evans (2010, Theorem 1 on page 275). □

Remark 2.1.9. *On convex domains $\Omega \subset \mathbb{R}^n$ in any dimension $n \in \mathbb{N}$, it holds $C_{\mathrm{P}}(\Omega) \leqslant 1/\pi$ (Payne and Weinberger, 1960; Bebendorf, 2003). For triangular domains $T \subset \mathbb{R}^2$, Laugesen and Siudeja (2010) recently showed the refined result $C_{\mathrm{P}}(T) \leqslant 1/j_{1,1}$ with the first positive root $j_{1,1}$ of the Bessel function J_1.*

Theorem 2.1.10 (Friedrichs Inequality)**.** *Let Ω be a Lipschitz domain and $\Gamma \subset \partial\Omega$ with nonzero $n-1$ dimensional Hausdorff measure. There exists a constant $C_{\mathrm{F}}(\Omega, \Gamma)$ such that, for any function $v \in H^1(\Omega)$ with $v = 0$ along Γ, it holds*

$$\|v\|_{L^2(\Omega)} \leqslant C_{\mathrm{F}}(\Omega, \Gamma) \operatorname{diam}(\Omega) \|\nabla v\|_{L^2(\Omega)}.$$

The constant $C_{\mathrm{F}}(\Omega, \Gamma)$ is invariant under rescaling of Ω.

Proof. A proof can be found in Braess (2007, page 30). □

Theorem 2.1.11 (Trace Theorem)**.** *Let Ω be a Lipschitz domain and $\Gamma \subset \partial\Omega$ with nonzero $n-1$ dimensional Hausdorff measure. Then there exists a constant $C_{\mathrm{T}}(\Omega)$, such that, for any function $v \in H^1(\Omega)$ with $v = 0$ along Γ, it holds*

$$\|v\|_{L^2(\partial\Omega)} \leqslant C_{\mathrm{T}}(\Omega, \Gamma) \operatorname{diam}(\Omega)^{1/2} \|\nabla v\|_{L^2(\Omega)}.$$

The constant $C_{\mathrm{T}}(\Omega, \Gamma)$ is invariant under rescaling of Ω.

Proof. Brenner and Scott (2008, Theorem 1.6.6 on page 39) prove

$$\|v\|_{L^2(\partial\Omega)} \leqslant C \|v\|^{1/2}_{L^2(\Omega)} \|\nabla v\|^{1/2}_{L^2(\Omega)}.$$

A Friedrichs inequality estimates $\|v\|^{1/2}_{L^2(\Omega)}$ on the right-hand side and concludes the proof with $C_{\mathrm{T}}(\Omega, \Gamma) \coloneqq C C_{\mathrm{F}}(\Omega, \Gamma)^{1/2}$. □

2.1.4 Helmholtz Decomposition

The following theorem establishes the Helmholtz decomposition of vector fields into some gradient and some Curl defined as

$$\begin{aligned} \operatorname{Curl} v &:= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla v = \begin{pmatrix} -\partial v/\partial x_2 \\ \partial v/\partial x_1 \end{pmatrix} \qquad \text{for } n = 2 \text{ and } v \in H^1(\Omega), \\ \operatorname{Curl} v &:= \nabla \times v = \begin{pmatrix} \partial v_3/\partial x_2 - \partial v_2/\partial x_3 \\ \partial v_1/\partial x_3 - \partial v_3/\partial x_1 \\ \partial v_2/\partial x_1 - \partial v_1/\partial x_2 \end{pmatrix} \quad \text{for } n = 3 \text{ and } v \in H^1(\Omega;\mathbb{R}^3). \end{aligned} \tag{2.1}$$

To handle $n = 2, 3$ simultaneously, assume $s := 1$ for $n = 2$ and $s := 3$ for $n = 3$. For the following theorem, recall Definition 2.1.7 of $H^1_D(\Omega)$ and recall that ν denotes the outer unit normal vector of Ω along $\partial\Omega$.

Theorem 2.1.12 (Helmholtz Decomposition)**.** *Let Ω be a bounded (simply connected) Lipschitz domain. Given any $p \in L^2(\Omega;\mathbb{R}^n)$ and a symmetric, uniformly positive tensor $\mathbb{S} \in L^\infty(\Omega;\mathbb{R}^{n\times n})$, there exist $\alpha \in H^1_D(\Omega)$ and $\beta \in H^1(\Omega;\mathbb{R}^s)$ with $\operatorname{Curl}\beta \cdot \nu = 0$ on Γ_N such that*

$$\mathbb{S}p = \mathbb{S}\nabla\alpha + \operatorname{Curl}\beta.$$

This split is orthogonal in the sense that

(a) $\int_\Omega (\mathbb{S}p)\cdot\nabla v\,\mathrm{d}x = \int_\Omega (\mathbb{S}\nabla\alpha)\cdot\nabla v\,\mathrm{d}x$ *for all* $v \in H^1_D(\Omega)$
(b) $\int_\Omega p\cdot\operatorname{Curl} w\,\mathrm{d}x = \int_\Omega (\mathbb{S}^{-1}\operatorname{Curl}\beta)\cdot\operatorname{Curl} w\,\mathrm{d}x$ *for all* $w \in H^1(\Omega;\mathbb{R}^s)$.

Proof. The Lax-Milgram theory yields a unique solution $\alpha \in H^1_D(\Omega)$ for (a) with $\mathbb{S}\nabla\alpha\cdot\nu = \mathbb{S}p\cdot\nu$ along Γ_N. The remainder $q := \mathbb{S}(p - \nabla\alpha)$ is divergence-free, i.e. $q \in H(\operatorname{div},\Omega)$ with $\operatorname{div} q = 0$, and satisfies $q\cdot\nu = 0$ along Γ_N. Standard results in higher analysis, such as Theorem 3.1 (for $n = 2$) and Theorem 3.4 (for $n = 3$) from Girault and Raviart (1986) ensure the existence of some $\beta \in H^1(\Omega;\mathbb{R}^s)$ with $q = \operatorname{Curl}\beta$. An integration by parts and a density argument yields (b). □

2.2 Finite Element Spaces

Weak formulations of the model problems in this work lead to the test function spaces $H^1_0(\Omega)$ and $H(\operatorname{div},\Omega)$ as well as $L^2(\Omega)$. Finite element methods discretise these spaces and employ generalised splines on subdomains, i.e. triangles or tetrahedra. This section introduces suitable ansatz functions for a single subdomain, while the next section introduces regular triangulations which connect these subdomains and allow for global interpolation on the whole domain.

2.2.1 Finite Elements in the Sense of Ciarlet

The following introduction to finite elements follows the outline in Brenner and Scott (2008) and the basic concepts of Ciarlet (1978).

Definition 2.2.1 (Finite Element in the Sense of Ciarlet (1978))**.** *The triple $(T,\mathcal{P},L)$ defines a finite element in the sense of Ciarlet if*

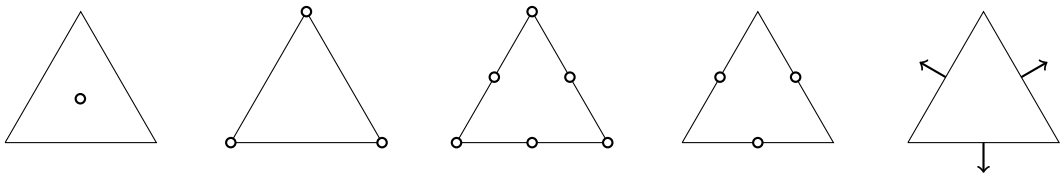

Figure 2.1: Lagrange (1-3), Crouzeix-Raviart (4) and Raviart-Thomas finite elements (5).

1. *$T \subset \mathbb{R}^n$ is a bounded closed set with nonempty interior and piecewise smooth boundary,*
2. *$\mathcal{P}$ is a finite-dimensional space of functions on T,*
3. *$L = \{L_1, L_2, \cdots, L_N\}$ is a basis of $\mathcal{P}^\star$.*

2.2.2 Lagrange, Crouzeix-Raviart and Raviart-Thomas Finite Elements

This subsection introduces the three best-known finite element classes illustrated in Figure 2.1. The markings in this figure relate to the linear functionals $L_1, L_2, \ldots, L_N$ or the degrees of freedom of the finite elements. In the following, δ_{jk} refers to the Kronecker delta, i.e., $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ otherwise.

Definition 2.2.2 (Local Polynomial Spaces)**.** *Given some triangle (for $n = 2$) or tetrahedron (for $n = 3$) T, the polynomials of degree k are denoted by $\mathcal{P}_k(T)$. The set of* Raviart-Thomas functions *of order k on T reads*

$$\mathrm{RT}_k(T) \coloneqq \left\{ v \in \mathcal{P}_{k+1}(T; \mathbb{R}^n) \,\middle|\, \exists a_0, a_1, \ldots, a_n \in P_k(T)\, \forall x \in T,\, v(x) = (a_1, \ldots, a_n) + a_0 x \right\}.$$

Definition 2.2.3 (Local Nodal Basis Function)**.** *Given some triangle (for $n = 2$) or tetrahedron (for $n = 3$) $T = \mathrm{conv}\{P_1, \ldots, P_{n+1}\}$, the* local nodal basis function *φ_{P_j} of the node P_j is defined by*

$$\varphi_{P_j} \in \mathcal{P}_1(T) \qquad \text{and} \qquad \varphi_{P_j}(P_k) = \delta_{jk} \quad \text{for all } k = 1, \ldots, n+1.$$

Remark 2.2.4. *The nodal basis functions $\varphi_{P_1}, \ldots, \varphi_{P_{n+1}}$ coincide with the barycentric coordinates in the sense that, for every $x \in T$, it holds*

$$x = \sum_{j=1}^{n+1} \varphi_{P_j}(x) P_j \qquad \text{and} \qquad \sum_{j=1}^{n+1} \varphi_{P_j} \equiv 1.$$

Furthermore, they define a basis of $\mathcal{P}_1(T)$.

Theorem 2.2.5 (Lagrange Finite Elements)**.** *Let $T = \mathrm{conv}\{P_1, \ldots, P_{n+1}\}$ be some triangle (for $n = 2$) or tetrahedron (for $n = 3$) with center $\mathrm{mid}(T)$ and sides $E_1, \ldots, E_{n+1}$ enumerated as shown in Figure 2.2, i.e. $E_j \coloneqq \mathrm{conv}\{P_k \,|\, k = 1, \ldots, n+1 \;\&\; k \neq j\}$. The points $R_{jk} \coloneqq (P_j + P_k)/2$ denote the edge midpoints for $1 \leqslant j < k \leqslant n+1$ (and coincide with the centers of $E_1, \ldots, E_3$ for $n = 2$). Furthermore, consider the point evaluations $\varrho_x(v) \coloneqq v(x)$ for $v \in \mathcal{P}_2(T)$ and any $x \in T$.*

(a) The triple $(T, \mathcal{P}_0(T), (\varrho_{\mathrm{mid}(T)}))$ defines a finite element in the sense of Ciarlet.

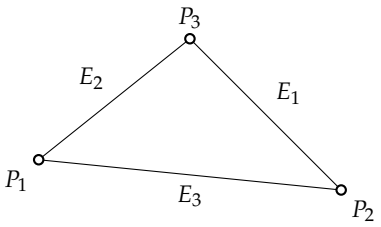

Figure 2.2: Standard enumeration of vertices and faces in a triangle. The vertex P_j is opposite to the face E_j.

(b) The triple $(T, \mathcal{P}_1(T), (\varrho_{P_1}, \dots, \varrho_{P_{n+1}}))$ defines a finite element in the sense of Ciarlet.
(c) The triple $(T, \mathcal{P}_2(T), (\varrho_{P_1}, \dots, \varrho_{P_{n+1}}, \varrho_{R_{jk}}$ for $1 \leqslant j < k \leqslant n+1))$ defines a finite element in the sense of Ciarlet.

Proof. Assertion (a) is trivial. The barycentric coordinates $\varphi_{P_1}, \dots, \varphi_{P_{n+1}}$ define a basis of $\mathcal{P}_1(T)$. Since $\varrho_{P_j}(\varphi_{P_k}) = \delta_{jk}$ and $\dim \mathcal{P}_1 = \dim \mathcal{P}_1^\star = n+1$, the functionals $\varrho_{P_1}, \dots, \varrho_{P_{n+1}}$ define a basis of $\mathcal{P}_1(T)^\star$. This proves (b). For (c) and $\mathcal{P}_2(T)$ with dimension $\dim \mathcal{P}_2(T) = (n+2)(n+1)/2$, consider the same number of functions

$$\hat{\varphi}_{P_j} = \varphi_j(2\varphi_j - 1) \quad \text{for } j = 1, \dots, n+1 \quad \text{and}$$
$$\hat{\varphi}_{R_{jk}} = 4\varphi_{P_j}\varphi_{P_k} \quad \text{for } 1 \leqslant j < k \leqslant n+1.$$

These functions satisfy $\varrho_X(\hat{\varphi}_Y) = \delta_{XY}$ for any $X, Y \in \{P_1, P_2, P_3, R_{jk} \text{ for } 1 \leqslant j < k \leqslant n+1\}$ and therefore are a valid dual basis. □

Theorem 2.2.6 (Crouzeix-Raviart Finite Element)**.** *Let $T = \operatorname{conv}\{P_1, \dots, P_{n+1}\}$ be some triangle (for $n = 2$) or tetrahedron (for $n = 3$) with sides $E_1, \dots, E_{n+1}$ as in Theorem 2.2.5 with associated linear functionals*

$$L_j(v) := \fint_{E_j} v \, \mathrm{d}s \qquad \text{for } j = 1, \dots, n+1 \text{ and } v \in \mathcal{P}_1(T)$$

The Crouzeix-Raviart basis functions

$$\psi_j(x) := 1 - n\varphi_j \quad \text{for } j = 1, \dots, n+1$$

form a dual basis for $(L_1, \dots, L_{n+1})$ and so the triple $(T, \mathcal{P}_1(T), (L_1, \dots, L_{n+1}))$ defines a finite element in the sense of Ciarlet.

Proof. By linearity, it follows $\varphi_j(\operatorname{mid}(E_k)) = (1 - \delta_{jk})/n$ and the Crouzeix-Raviart basis functions satisfy $\psi_j(\operatorname{mid}(E_k)) = \delta_{jk}$ and so form a dual basis of $\{L_1, \dots, L_{n+1}\}$. □

Theorem 2.2.7 (Raviart-Thomas Finite Element)**.** *Let $T = \operatorname{conv}\{P_1, \dots, P_{n+1}\}$ be some triangle (for $n = 2$) or tetrahedron (for $n = 3$) with sides $E_1, \dots, E_{n+1}$ as in Theorem 2.2.5 with associated linear functionals*

$$L_j(v) := \int_{E_j} v \cdot \nu_T|_{E_j} \, \mathrm{d}s \qquad \text{for } j = 1, \dots, n+1 \text{ and } v \in \mathrm{RT}_0(T)$$

where ν_T denotes the outer unit normal vector of T along ∂T. These functionals and the Raviart-Thomas basis functions

$$\vartheta_j(x) \coloneqq \frac{1}{n\,|T|}\,(x - P_j) \qquad \text{for } j = 1, \dots, n+1.$$

satisfy

(a) $L_j(\vartheta_k) = \delta_{jk}$, and $\vartheta_k(x) \cdot \nu_T|_{E_j} = \delta_{jk} / \left|E_j\right|$ for $x \in E_j$,
(b) the set $\{\vartheta_1, \dots, \vartheta_{n+1}\}$ is a basis of $\mathrm{RT}_0(T)$, and
(c) the triple $(T, \mathrm{RT}_0(T), (L_1, L_2, \dots, L_{n+1}))$ defines a finite element in the sense of Ciarlet.

Proof. Easy geometrical considerations show $(x - P_j) \cdot \nu_T|_{E_j} = n\,|T| \,/\, \left|E_j\right|$ for $x \in E_j$ and $(x - P_k) \cdot \nu_T|_{E_j} = 0$ for $x \in E_j$ and $k \neq j$. This proves (a). For the proof of (b) it is easy to check that $\vartheta_j \in \mathrm{RT}_0(T)$ for $j = 1, \dots, n+1$ and their linear independency follows from (a). Since, $\dim(\mathrm{RT}_0(T)) = n + 1$ this proves (b). The last assertion (c) follows directly from (a) and (b) similar to the theorems above. □

2.2.3 Regular Triangulations and Related Notation

Triangulations are the main tool for domain discretisation in finite element methods and are the topic of this subsection.

Definition 2.2.8 (Triangulation)**.** *A* triangulation $\mathcal{T}$ *of some polyhedral Lipschitz domain Ω consists of n-dimensional closed triangles (for $n = 2$) or tetrahedra (for $n = 3$) $T \in \mathcal{T}$, called elements, such that $\bigcup_{T \in \mathcal{T}} = \overline{\Omega}$. The set of all vertices in the triangulation is denoted by $\mathcal{N}$ and the set of all sides (edges for $n = 2$ and faces for $n = 3$) in the triangulation is denoted by $\mathcal{E}$. The subsets $\mathcal{E}(\partial\Omega) \coloneqq \{E \in \mathcal{E} \,\big|\, E \subseteq \partial\Omega\}$ contains all sides along the boundary $\partial\Omega$, while $\mathcal{E}(\Omega) \coloneqq \mathcal{E} \backslash \mathcal{E}(\partial\Omega)$ contains all interior sides.*

Definition 2.2.9 (Regular Triangulation)**.** *A triangulation is called regular if the intersection of any two elements $T_1, T_2 \in \mathcal{T}$ with $T_1 \neq T_2$ equals $T_1 \cap T_2 = \varnothing$ or*

$$T_1 \cap T_2 = \mathrm{conv}\{z_1, \dots, z_k\} \quad \text{for } 1 \leqslant k \leqslant n \text{ nodes } z_1, \dots, z_k \in \mathcal{N}.$$

(Hence, the intersection equals a single node, an edge or a face of the triangulation.) For mixed boundary conditions there is the additional condition that the intersection of any boundary edge $E \in \mathcal{E}(\partial\Omega)$ with Γ_D satisfies

$$E \cap \Gamma_D = E \qquad \text{or} \qquad E \cap \Gamma_D = \varnothing.$$

Figure 2.3 displays some examples for triangulations that satisfy or violate these conditions.

To simplify the presentation in the rest of this thesis some notation is in order. First, there are several local subsets of $\mathcal{E}$, $\mathcal{T}$ and $\mathcal{N}$, namely

$$\begin{aligned}
\mathcal{N}(E) &\coloneqq \{z \in \mathcal{N} \,\big|\, z \in E\} && \text{and} && \mathcal{T}(E) \coloneqq \{T \in \mathcal{T} \,\big|\, E \subset \partial T\} && \text{for } E \in \mathcal{E}, \\
\mathcal{N}(T) &\coloneqq \{z \in \mathcal{N} \,\big|\, z \in T\} && \text{and} && \mathcal{E}(T) \coloneqq \{E \in \mathcal{E} \,\big|\, E \subset \partial T\} && \text{for } T \in \mathcal{T}, \\
\mathcal{E}(z) &\coloneqq \{E \in \mathcal{E} \,\big|\, z \in E\} && \text{and} && \mathcal{T}(z) \coloneqq \{T \in \mathcal{T} \,\big|\, z \in T\} && \text{for } z \in \mathcal{N}.
\end{aligned}$$

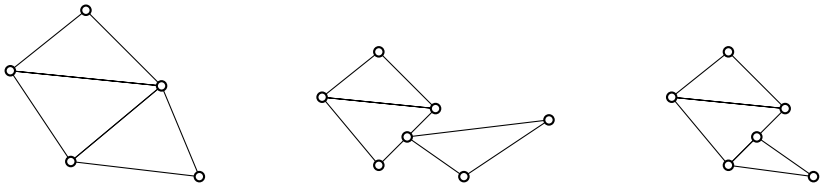

Figure 2.3: One regular (left) and two non-regular (middle and right) triangulations.

Second, there are different types of patches that are

$$\begin{aligned} \text{the node patch} \quad \omega_z &:= \operatorname{int}\Big(\bigcup \mathcal{T}(z)\Big) \quad \text{for } z \in \mathcal{N}, \\ \text{the side patch} \quad \omega_E &:= \operatorname{int}\Big(\bigcup \mathcal{T}(E)\Big) \quad \text{for } E \in \mathcal{E}, \\ \text{the element patch} \quad \omega_T &:= \bigcup_{z \in \mathcal{N}(T)} \omega_z \quad \text{for } T \in \mathcal{T}. \end{aligned}$$

Third, the boundary sides $\mathcal{E}(\partial\Omega)$ split into the Dirichlet and Neumann sides

$$\mathcal{E}(\Gamma_D) := \{E \in \mathcal{E}(\partial\Omega) \,\big|\, E \subseteq \Gamma_D\} \quad \text{and} \quad \mathcal{E}(\Gamma_N) := \mathcal{E}(\partial\Omega)\backslash\mathcal{E}(\Gamma_D).$$

The associated sets of triangles read

$$\mathcal{T}(\Gamma_D) := \{T \in \mathcal{T} \,\big|\, \mathcal{E}(T) \cap \mathcal{E}(\Gamma_D) \neq \varnothing\} \quad \text{and} \quad \mathcal{T}(\Gamma_N) := \{T \in \mathcal{T} \,\big|\, \mathcal{E}(T) \cap \mathcal{E}(\Gamma_N) \neq \varnothing\}.$$

Similarly, there there are Dirichlet boundary nodes

$$\mathcal{N}(\Gamma_D) := \{z \in \mathcal{N} \,\big|\, z \in \Gamma_D\}.$$

The remaining nodes form the set of free nodes,

$$\mathcal{M} := \mathcal{N}\backslash\mathcal{N}(\Gamma_D).$$

2.2.4 Interpolation Operators and Finite Element Spaces

This subsection applies the results on finite elements of Subsection 2.2.2 to each element of a regular triangulation and so defines interpolation operators on Ω. Furthermore this subsection defines finite element approximation spaces as the codomains of these operators. Such approximation spaces are required for the finite element methods of Section 2.3.

Definition 2.2.10 (Local Interpolant)**.** *Let $(T, \mathcal{P}, L)$ be some finite element with dual basis $\{\xi_1, \ldots, \xi_N\}$ of L, i.e., $L_j(\xi_k) = \delta_{jk}$. Furthermore, let v be some function for that $L(v)$ is well-defined. Then, the* local interpolation *of v on T reads*

$$\mathcal{I}(v)|_T := \sum_{j=1}^{N} L_j(v)\xi_j.$$

The $\mathcal{T}$-piecewise application of the local interpolant to a function in $C^\infty(\overline{\Omega})$ leads to an interpolant that is defined on Ω. This global interpolant is broken in the sense that it is possibly discontinuous along sides of the triangulation. However, depending on the structure of the finite element, there are some continuity properties. The following definitions introduce broken polynomial spaces and global basis functions to formalise this.

Definition 2.2.11 (Broken Polynomial Spaces)**.** *For a collection $\mathcal{T}$ of elements, the* set of piecewise polynomials *on $\bigcup \mathcal{T} \coloneqq \bigcup_{T\in\mathcal{T}} T$ reads*

$$\mathcal{P}_k(\mathcal{T}) \coloneqq \left\{ v \in L^2\left(\bigcup \mathcal{T}\right) \mid \forall T \in \mathcal{T},\ v|_T \in P_1(\mathcal{T}) \right\}.$$

The broken Raviart-Thomas *space of lowest order $k = 0$ reads*

$$\mathrm{RT}_{-1}(\mathcal{T}) \coloneqq \left\{ v \in L^2\left(\bigcup \mathcal{T}; \mathbb{R}^n\right) \mid \forall T \in \mathcal{T},\ v|_T \in \mathrm{RT}_0(T) \right\}.$$

Definition 2.2.12 (Global Basis Functions)**.** *For a given regular triangulation $\mathcal{T}$, the* nodal basis function *φ_z for some node $z \in \mathcal{N}$ is given by*

$$\varphi_z \in \mathcal{P}_1(\mathcal{T}), \quad \textit{and} \quad \varphi_z(y) \coloneqq \begin{cases} 1 & \textit{for } z = y, \\ 0 & \textit{for } z \neq y \in \mathcal{N}. \end{cases}$$

Similarly, the Crouzeix-Raviart basis function *ψ_E for some side $E \in \mathcal{E}$ is given by*

$$\psi_E \in \mathcal{P}_1(\mathcal{T}), \quad \textit{and} \quad \psi_E(\mathrm{mid}(F)) \coloneqq \begin{cases} 1 & \textit{for } E = F, \\ 0 & \textit{for } E \neq F \in \mathcal{E}. \end{cases}$$

Locally, these global basis functions coincide with the local basis functions of Theorems 2.2.5 and 2.2.6.

Finally, the Raviart-Thomas basis function *$\vartheta_E \in \mathrm{RT}_{-1}(\mathcal{T})$ for some side $E \in \mathcal{E}$ with some arbitrary but fixed oriented normal vector ν_E is given by*

$$\vartheta_E(x)|_T \coloneqq (\nu_E \cdot \nu_T) \frac{1}{n\,|T|}\,(x - P) \qquad \textit{in } x \in T = \mathrm{conv}\{E, P\} \in \mathcal{T}(E).$$

Furthermore $\vartheta \equiv 0$ on $T \in \mathcal{T}\backslash\mathcal{T}(E)$. Locally, the function ϑ_E coincides with the local basis functions of Theorem 2.2.7 on $T \in \mathcal{T}(E)$ up to the sign $\nu_T \cdot \nu_E \in \{-1, 1\}$.

Remark 2.2.13. *Subsubsection 2.3.5.5 below explains how the orientation is fixed in the implementation for this thesis.*

Notice that φ_z is continuous in Ω and that ψ_E is continuous in the midpoint $\mathrm{mid}(E)$ of the side $E \in \mathcal{E}$. The function $\vartheta_E \cdot \nu_E$ is continuous along the side $E \in \mathcal{E}$. These continuity properties are reflected in the following definitions of the interpolation operators and their codomains.

Definition 2.2.14 (Nodal Interpolation and $\mathcal{P}_1$ Conforming Finite Element Space)**.** *The application of the local interpolant of the $\mathcal{P}_1$ Lagrange finite element with the nodal basis functions*

φ_z from Definition 2.2.12 on each element $T \in \mathcal{T}$ of some regular triangulation $\mathcal{T}$ leads to the nodal interpolation

$$\mathcal{I}_{\mathcal{N}} : C(\Omega) \to \mathcal{P}_1(\mathcal{T}) \cap C(\Omega), \qquad \mathcal{I}_{\mathcal{N}}(v) \coloneqq \sum_{z \in \mathcal{N}} v(z)\varphi_z$$

and to the $\mathcal{P}_1$ conforming finite element space

$$\mathcal{P}_1(\mathcal{T}) \cap C(\Omega) = \mathcal{I}_{\mathcal{N}}(C(\Omega)).$$

Definition 2.2.15 (Nonconforming Interpolation and Crouzeix-Raviart Finite Element Space)**.** *The application of the local interpolant of the Crouzeix-Raviart finite element with the basis functions ψ_E from Definition 2.2.12 on each element $T \in \mathcal{T}$ of some regular triangulation $\mathcal{T}$ leads to the* nonconforming interpolation

$$\mathcal{I}_{\mathrm{NC}} : H^1(\Omega) \to \mathrm{CR}(\mathcal{T}), \qquad \mathcal{I}_{\mathrm{NC}}(v) \coloneqq \sum_{E \in \mathcal{E}} \left(\fint_E v \,\mathrm{d}s \right) \psi_E$$

and to the Crouzeix-Raviart nonconforming finite element space

$$\mathrm{CR}(\mathcal{T}) \coloneqq \mathcal{P}_1(\mathcal{T}) \cap C(\mathrm{mid}(\mathcal{E})) = \mathcal{I}_{\mathrm{NC}}(H^1(\Omega)).$$

Definition 2.2.16 (Fortin Interpolation and Raviart-Thomas Finite Element Space)**.** *The application of the local interpolant of the lowest-order Raviart-Thomas finite element with the basis functions ϑ_E and their oriented normal vectors ν_E from Definition 2.2.12 on each element $T \in \mathcal{T}$ leads to the* Fortin interpolation

$$\mathcal{I}_{\mathrm{RT}} : \big(H(\mathrm{div}, \Omega) \cap L^{2+\varepsilon}(\Omega)\big) \to \mathrm{RT}_0(\mathcal{T}), \qquad \mathcal{I}_{\mathrm{RT}}(q) \coloneqq \sum_{E \in \mathcal{E}} \left(\int_E q \cdot \nu_E \,\mathrm{d}s \right) \vartheta_E$$

and to the Raviart-Thomas finite element space

$$\mathrm{RT}_0(\mathcal{T}) \coloneqq \big\{ q \in \mathrm{RT}_{-1}(\mathcal{T}) \,\big|\, \forall E \in \mathcal{E}(\Omega),\ q \cdot \nu_E \in C(E) \big\} = \mathcal{I}_{\mathrm{RT}}(H(\mathrm{div}, \Omega) \cap L^{2+\varepsilon}(\Omega)).$$

The additional constraint $q \in L^{2+\varepsilon}(\Omega)$ for some $\varepsilon > 0$ implies that the integral $\int_E q \cdot \nu_E \,\mathrm{d}s$ exists for any $E \in \mathcal{E}$, cf. Brezzi and Fortin (1991, Section III 3.3, p. 125) for an explanation on this.

2.2.5 Useful Identities

The first identity is the point of departure for many explicit estimates connected to traces of Sobolev functions along the sides $\mathcal{E}$ of a triangulation $\mathcal{T}$.

Lemma 2.2.17 (Trace Identity)**.** *Given a function $w \in H^1(T)$ on an element $T = \mathrm{conv}\,\{E, \{P\}\}$ with a side $E \in \mathcal{E}(T)$ and its opposite vertex P, it holds*

$$\fint_E w \,\mathrm{d}s = \fint_T w \,\mathrm{d}x + \frac{1}{n} \fint_T (x - P) \cdot \nabla w \,\mathrm{d}x.$$

Proof. The proof employs Theorem 2.1.5.(b) with $\mathrm{div}(x - P) = n$ and the fact that $(x -$

$P) \cdot \nu_T|_E = n\,|T|\,/\,|E|$ for $x \in E$ and $(x - P) \cdot \nu_T = 0$ for $x \in \partial T \backslash E$. □

The next formula is a very handy tool to compute the entries of mass matrices that involve powers of nodal basis functions.

Lemma 2.2.18. *For an element* $T = \operatorname{conv}\{P_1, \ldots, P_{n+1}\}$ *with nodal basis functions* $\varphi_{P_1}, \ldots, \varphi_{P_{n+1}}$ *and* $\alpha_1, \ldots, \alpha_{n+1} \in \mathbb{N}_0$*, it holds*

$$\int_T \varphi_{P_1}^{\alpha_1} \cdots \varphi_{P_{n+1}}^{\alpha_{n+1}} \,\mathrm{d}x = n!\,|T|\ \frac{\alpha_1! \cdots \alpha_{n+1}!}{(n + \alpha_1 + \ldots + \alpha_{n+1})!}.$$

Proof. The factor $n!\,|T|$ stems from the transformation to the reference tetrahedron $T_{\mathrm{ref}} \coloneqq \operatorname{conv}\{0, e_1, \ldots, e_n\}$ where e_j is the unit vector in the j-th coordinate direction. In fact, it is the determinant of the gradient of the affine transformation

$$B : T_{\mathrm{ref}} \to T, \qquad B(x) = P_1 + x\left((P_2 - P_1)^{\mathrm{T}}, (P_3 - P_1)^{\mathrm{T}}, \ldots, (P_{n+1} - P_1)^{\mathrm{T}}\right).$$

Note that x and P_j are treated as row vectors. On T_{ref} the nodal basis functions read

$$\varphi_{\mathrm{ref},1}(x) \coloneqq 1 - \sum_{j=1}^{n} x_j, \quad \text{and} \quad \varphi_{\mathrm{ref},j+1}(x) \coloneqq x_j \quad \text{for } j = 1, \ldots, n.$$

An iterated integration by parts shows, for any $k \in \{1, \ldots n\}$, fixed $\gamma, \beta \in \mathbb{N}_0$ and fixed coordinates $x_{k+1}, \ldots, x_n$,

$$\begin{aligned}\int_0^{1-\sum_{j=k+1}^n x_j} x_k^\beta (1 - \sum_{j=k}^{n} x_j)^\gamma \,\mathrm{d}x_k &= \frac{\gamma}{\beta+1} \int_0^{1-\sum_{j=k+1}^n x_j} x_k^{\beta+1} (1 - \sum_{j=k}^{n} x_j)^{\gamma-1} \,\mathrm{d}x_k \\ &= \ldots = \frac{\beta!\gamma!}{(\beta+\gamma)!} \int_0^{1-\sum_{j=k+1}^n x_j} x_k^{\beta+\gamma} \,\mathrm{d}x_k \\ &= \frac{\beta!\gamma!}{(\beta+\gamma+1)!} (1 - \sum_{j=k+1}^{n} x_j)^{\beta+\gamma+1}.\end{aligned}$$

Successive utilisation of this formula for $k = 1, \ldots, n$, beginning with the innermost integral, yields

$$\int_{T_{\mathrm{ref}}} \varphi_{\mathrm{ref},1}^{\alpha_1} \cdot \cdots \cdot \varphi_{\mathrm{ref},n+1}^{\alpha_{n+1}} \,\mathrm{d}x$$

$$
\begin{aligned}
&= \int_0^1 \int_0^{1-x_n} \cdots \int_0^{1-\sum_{j=2}^n x_j} x_1^{\alpha_1} \cdots x_n^{\alpha_n} (1 - \sum_{j=1}^n x_j)^{\alpha_{n+1}} \,\mathrm{d}x_1 \ldots \,\mathrm{d}x_{n-1} \,\mathrm{d}x_n \\
&= \frac{\alpha_1! \alpha_{n+1}!}{(\alpha_1 + \alpha_{n+1} + 1)!} \\
&\quad \times \int_0^1 \int_0^{1-x_n} \cdots \int_0^{1-\sum_{j=3}^n x_j} x_2^{\alpha_2} \cdots x_n^{\alpha_n} (1 - \sum_{j=2}^n x_j)^{\alpha_1 + \alpha_{n+1} + 1} \,\mathrm{d}x_2 \ldots \,\mathrm{d}x_{n-1} \,\mathrm{d}x_n \\
&= \ldots = \frac{\alpha_1! \cdots \alpha_{n+1}!}{(n + \alpha_1 + \ldots + \alpha_{n+1})!}.
\end{aligned}
$$

This completes the proof. □

The following lemma provides an explicit formula for a term that occurs frequently in estimates that involve the trace identity (Theorem 2.2.17).

Lemma 2.2.19. *For an element* $T = \operatorname{conv}\{\{P\} \cup E\} = \operatorname{conv}\{P, P_2, \ldots, P_{n+1}\}$ *with side* $E \in \mathcal{E}(T)$*, its opposite vertex* $P = P_1 \in \mathcal{N}(T)$ *and* $s(T)^2 := \sum_{j,k=1}^{n+1} |P_j - P_k|^2 / 2$*, it holds*

$$\|\bullet - \operatorname{mid}(T)\|_{L^2(T)}^2 = |T| \, \frac{s(T)^2}{(n+2)(n+1)^2}.$$

Proof. Let $\varphi_1, \ldots, \varphi_{n+1}$ denote the nodal basis functions of the element T with the vertices $P_1, \ldots, P_{n+1}$. Then

$$x = \varphi_1(x) P_1 + \ldots + \varphi_{n+1}(x) P_{n+1} \quad \text{for } x \in T.$$

Without loss of generality, let $0 = \operatorname{mid}(T) = P_1 + P_2 + \ldots + P_{n+1}$. Hence,

$$\int_T |x|^2 \,\mathrm{d}x = \sum_{j,k=1}^{n+1} P_j \cdot P_k \int_T \varphi_j \varphi_k \,\mathrm{d}x = \sum_{j,k=1}^{n+1} P_j \cdot P_k (1 + \delta_{jk}) \, |T| \, n!/(n+2)!.$$

Since

$$0 = |P_1 + P_2 + \ldots + P_{n+1}|^2 = \sum_{j,k=1}^{n+1} P_j \cdot P_k,$$

it follows

$$\int_T |x|^2 \,\mathrm{d}x = |T| \, n!/(n+2)! \sum_{j=1}^{n+1} |P_j|^2.$$

The same orthogonality shows

$$2s(T)^2 = \sum_{j,k=1}^{n+1} |P_j - P_k|^2 = \sum_{j,k=1}^{n+1} \Big(|P_j|^2 + |P_k|^2 - 2 P_j \cdot P_k \Big) = 2(n+1) \sum_{j=1}^{n+1} |P_j|^2.$$

The combination of the last two equations concludes the proof. □

Theorem 2.2.20 (Discrete Helmholtz decomposition)**.** *For a regular triangulation $\mathcal{T}$ of some simply connected domain $\Omega \subset \mathbb{R}^2$, it holds*

$$\mathcal{P}_0(\mathcal{T};\mathbb{R}^2) = \operatorname{Curl}\Big(\big(\mathcal{P}_1(\mathcal{T}) \cap C(\Omega)\big)/\mathbb{R}\Big) \oplus \nabla_{\mathrm{NC}}\Big(\mathrm{CR}_0(\mathcal{T})\Big)$$

and the direct sum is orthogonal in $L^2(\Omega;\mathbb{R}^2)$.

Proof. The key arguments of the proof are the following. The dimensions of the involved function spaces are

$$\begin{aligned} \dim(\mathcal{P}_0(\mathcal{T};\mathbb{R}^2)) &= 2\,|\mathcal{T}|\,, \\ \dim\Big(\big(\mathcal{P}_1(\mathcal{T}) \cap C(\Omega)\big)/\mathbb{R}\Big) &= |\mathcal{N}| - 1, \\ \dim(\mathrm{CR}_0(\mathcal{T})) &= |\mathcal{E}(\Omega)| \end{aligned}$$

and mathematical induction shows

$$2\,|\mathcal{T}| = |\mathcal{N}| + |\mathcal{E}(\Omega)| - 1.$$

Hence, it remains to prove the orthogonality

$$\int_\Omega \operatorname{Curl} \varphi_z \cdot \nabla_{\mathrm{NC}} \psi_E \,\mathrm{d}x = 0 \quad \text{for any } z \in \mathcal{N}, E \in \mathcal{E}(\Omega)$$

which follows from an integration by parts and properties of φ_z and ψ_E. A complete proof can be found in Arnold and Falk (1989) or Carstensen (2009). □

2.3 Finite Element Method

This section explains the finite element method for the Poisson model problem, based on the finite element spaces of Section 2.2.

2.3.1 Poisson Model Problem

For a polyhedral Lipschitz domain Ω with Dirichlet and Neumann boundary Γ_D and Γ_N as in Section 2.1.1, the strong formulation of the Poisson model problem for given data $f : \Omega \to \mathbb{R}$ and $g : \Gamma_N \to \mathbb{R}$ seeks $u \in C^2(\Omega)$ with

$$-\operatorname{div}(\mathbb{S}\nabla u) = f \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma_D \quad \text{and} \quad \nabla u \cdot \nu = g \text{ on } \Gamma_N.$$

The variational setting employs the bilinear form

$$a : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}, \quad a(u,v) \coloneqq \int_\Omega \mathbb{S}\nabla u \cdot \nabla v \,\mathrm{d}x.$$

The quantity $\mathbb{S} \in L^\infty(\Omega; \mathbb{R}^{n\times n})$ is assumed to be symmetric and uniformly positive definite with bounds $0 < \lambda_{\min,\Omega} \leqslant \lambda_{\max,\Omega} < \infty$ such that

$$\lambda_{\min,\Omega} \|\nabla v\|^2_{L^2(\Omega)} \leqslant a(v,v) \leqslant \lambda_{\max,\Omega} \|\nabla v\|^2_{L^2(\Omega)} \qquad \text{for all } v \in H^1(\Omega).$$

The tensor $\mathbb{S}$ describes different physical behavior in subdomains, e.g. for modelling of diffusion, dispersion or elasticity. The bilinear form a induces the energy norm

$$|||\cdot||| := a(\cdot,\cdot)^{1/2} = \left\|\mathbb{S}^{1/2}\nabla\cdot\right\|_{L^2(\Omega)}. \tag{2.2}$$

The right-hand side functional for the variational formulation reads

$$F : L^2(\Omega) \to \mathbb{R}, \quad F(v) := \int_\Omega fv \,\mathrm{d}x + \int_{\Gamma_N} gv \,\mathrm{d}s$$

with some volume force or source field $f \in L^2(\Omega)$ and Neumann boundary function $g \in L^2(\Gamma_N)$.

2.3.2 Primal Formulation and Discretisation

The primal weak formulation seeks $u \in V := H^1_D(\Omega) := \{v \in H^1(\Omega) \,|\, v = 0 \text{ on } \Gamma_D\}$ with

$$a(u,v) = F(v) \quad \text{for all } v \in V. \tag{2.3}$$

For $|\Gamma_D| > 0$, the Lax-Milgram theorem ensures unique solvability for the Hilbert space (V,a). For $\Gamma_D = \varnothing$, set $V := H^1(\Omega)/\mathbb{R}$. The finite element method replaces V by some discrete space V_h and computes $u_h \in V_h$ with

$$a(u_h,v_h) = F(v_h) \quad \text{for all } v_h \in V_h. \tag{2.4}$$

If $V_h \subseteq V$ (as for the P_1-conforming finite element space $V(\mathcal{T}) := P_1(\mathcal{T}) \cap V$), the method is called conform and leads immediately to the *Galerkin orthogonality*

$$a(u-u_h,v_h) = 0 \qquad \text{for all } v_h \in V_h. \tag{2.5}$$

This property makes u_h the Galerkin projection or best-approximation of u with respect to the energy norm $|||\cdot||| := a(\cdot,\cdot)^{1/2}$ induced by a. This is the statement of Lemma 2.3.1.(a) by Cea below.

Lemma 2.3.1. *For a solution $u \in V$ of the weak problem* (2.3) *and a discrete solution $u_h \in V_h \subset V$ of the discrete problem* (2.4) *with Galerkin orthogonality* (2.5)*, it holds*

(a) $|||u-u_h||| = \inf_{v_h\in V_h} |||u-v_h|||$ ***(Cea's Lemma)*** *,*
(b) $|||u-u_h|||^2 = |||u|||^2 - |||u_h|||^2$ *.*

Proof. Both assertions are an easy consequence of the Galerkin orthogonality (2.5). □

If otherwise $V_h \nsubseteq V$, as for the Crouzeix-Raviart nonconforming finite element space

$$V_h = \mathrm{CR}_0(\mathcal{T}) := \{v_{\mathrm{CR}} \in \mathrm{CR}(\mathcal{T}) \,|\, v_{\mathrm{CR}}(\mathrm{mid}(\mathcal{E}(\Gamma_D)) = 0)\},$$

the method is called nonconforming and replaces a by the extended mesh-dependent bilinear form

$$a_{\mathrm{NC}} : H^1(\mathcal{T}) \times H^1(\mathcal{T}) \to \mathbb{R}, \qquad a_{\mathrm{NC}}(u,v) \coloneqq \int_\Omega \mathbb{S}\nabla_{\mathcal{T}} u \cdot \nabla_{\mathcal{T}} v \,\mathrm{d}x \coloneqq \sum_{T\in\mathcal{T}} \int_T \mathbb{S}\nabla u|_T \cdot \nabla v|_T \,\mathrm{d}x \tag{2.6}$$

with energy norm $|||\cdot|||_{\mathrm{NC}} \coloneqq a_{\mathrm{NC}}(\cdot,\cdot)^{1/2}$ on the broken Sobolev space $H^1(\mathcal{T}) \coloneqq \{v \in L^2(\Omega) \,\big|\, v|_T \in H^1(T) \text{ for all } T \in \mathcal{T}\}$. The nonconforming finite element method for the Poisson model problem seeks $u_{\mathrm{NC}} \in V_h$ with

$$a_{\mathrm{NC}}(u_{\mathrm{NC}}, v_{\mathrm{NC}}) = F(v_{\mathrm{NC}}) \quad \text{for all } v_{\mathrm{NC}} \in V_h. \tag{2.7}$$

Since $a_{\mathrm{NC}}(u, v_{\mathrm{NC}}) \neq F(v_{\mathrm{NC}})$ in general, there is no Galerkin orthogonality. However, the second lemma of Strang provides a generalisation of Cea's Lemma.

Lemma 2.3.2 (Second Lemma by Berger, Scott and Strang)**.** *For a solution $u \in V$ of the weak problem* (2.3) *and a discrete nonconforming solution $u_{\mathrm{NC}} \in V_h$ of the discrete problem* (2.7)*, it holds*

$$|||u - u_{\mathrm{NC}}|||_{\mathrm{NC}} \leqslant 2 \inf_{v_{\mathrm{NC}}\in V_h} |||u - v_{\mathrm{NC}}|||_{\mathrm{NC}} + \sup_{w_{\mathrm{NC}}\in V_h} \frac{F(w_{\mathrm{NC}}) - a_{\mathrm{NC}}(u, w_{\mathrm{NC}})}{|||w_{\mathrm{NC}}|||_{\mathrm{NC}}}.$$

Remark 2.3.3. *The first term on the right-hand side of the Strang lemma is called the approximation error, while the second term is the additional consistency error of the nonconformity.*

Proof. The triangle inequality for any $v_{\mathrm{NC}} \in V_h$ yields

$$|||u - u_{\mathrm{NC}}|||_{\mathrm{NC}} \leqslant |||u - v_{\mathrm{NC}}|||_{\mathrm{NC}} + |||v_{\mathrm{NC}} - u_{\mathrm{NC}}|||_{\mathrm{NC}}.$$

The linearity of a_{NC} and (2.7) show, for $w_{\mathrm{NC}} = u_{\mathrm{NC}} - v_{\mathrm{NC}}$,

$$|||w_{\mathrm{NC}}|||_{\mathrm{NC}}^2 = a_{\mathrm{NC}}(w_{\mathrm{NC}}, w_{\mathrm{NC}}) = a_{\mathrm{NC}}(u - v_{\mathrm{NC}}, w_{\mathrm{NC}}) + F(w_{\mathrm{NC}}) - a_{\mathrm{NC}}(u, w_{\mathrm{NC}}).$$

A Hölder inequality in the first summand and division by $|||w_{\mathrm{NC}}|||_{\mathrm{NC}}$ gives

$$|||w_{\mathrm{NC}}|||_{\mathrm{NC}} \leqslant |||u - v_{\mathrm{NC}}|||_{\mathrm{NC}} + \frac{F(w_{\mathrm{NC}}) - a_{\mathrm{NC}}(u, w_{\mathrm{NC}})}{|||w_{\mathrm{NC}}|||_{\mathrm{NC}}}.$$

Since this holds for all $v_{\mathrm{NC}} \in V_h$, we can write the infimum in front of the first term. The supremum in the second term comes for free and concludes the proof. □

2.3.3 Dual Formulation and Discretisation

The dual formulation involves the space of $H(\mathrm{div}, \Omega)$ functions with homogeneous Neumann boundary conditions along Γ_N,

$$H_N(\mathrm{div}, \Omega) \coloneqq \left\{ q \in H(\mathrm{div}, \Omega) \,\big|\, q \cdot \nu = 0 \text{ along } \Gamma_N \right\},$$

and seeks $p \in H_N(\mathrm{div}, \Omega)$ with $p \cdot \nu = g$ along Γ_N and $u \times L^2(\Omega)$ with

$$\begin{aligned} \int_\Omega \mathbb{S}^{-1} q \cdot r \,\mathrm{d}x + \int_\Omega \mathrm{div}\, r u \,\mathrm{d}x = 0 \qquad & \text{for all } r \in H_N(\mathrm{div}, \Omega), \\ \int_\Omega \mathrm{div}\, q v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}s \qquad & \text{for all } v \in L^2(\Omega). \end{aligned}$$

The first equation is a weak formulation of the substitution $\mathbb{S}\nabla u = p$ and the second equation is a weak formulation of $-\mathrm{div}\, p = f$. The Raviart-Thomas mixed finite element method (MFEM) replaces $H_N(\mathrm{div}, \Omega)$ by $Q(\mathcal{T}) := \mathrm{RT}_0(\mathcal{T}) \cap H_N(\mathrm{div}, \Omega)$ and $L^2(\Omega)$ by $\mathcal{P}_0(\mathcal{T})$. The Neumann boundary conditions are approximated by the Raviart-Thomas interpolation (compare Definition 2.2.16)

$$p_{N,\mathrm{RT}} := \sum_{E \in \mathcal{E}(\Gamma_N)} \left(\int_E g \,\mathrm{d}s \right) \vartheta_E.$$

Then, the Raviart-Thomas MFEM seeks $p_{\mathrm{RT}} \in p_{N,\mathrm{RT}} + Q(\mathcal{T})$ and $u_0 \in \mathcal{P}_0(\mathcal{T})$ with

$$\begin{aligned} \int_\Omega \mathbb{S}^{-1} p_{\mathrm{RT}} \cdot q_{\mathrm{RT}} \,\mathrm{d}x + \int_\Omega \mathrm{div}\, q_{\mathrm{RT}} u_0 \,\mathrm{d}x = 0 \qquad & \text{for all } q_{\mathrm{RT}} \in Q(\mathcal{T}), \\ \int_\Omega \mathrm{div}\, p_{\mathrm{RT}} v_0 \,\mathrm{d}x = \int_\Omega f v_0 \,\mathrm{d}s \qquad & \text{for all } v_0 \in \mathcal{P}_0(\mathcal{T}). \end{aligned} \tag{2.8}$$

This linear system of equation computes the L^2 best-approximation of σ and also σ_h as stated in the following Lemma.

Lemma 2.3.4. *The Raviart-Thomas solution p_{RT} is the L^2 best-approximation of σ and of the stress tensor σ_h of the $\mathcal{P}_1$ conforming FEM in the sense that*

$$p_{\mathrm{RT}} = \operatorname*{argmin}_{\substack{q_{\mathrm{RT}} \in p_{N,\mathrm{RT}} + Q(\mathcal{T}) \\ \mathrm{div}\, q_{\mathrm{RT}} = -f_{\mathcal{T}}}} \left\| \mathbb{S}^{-1/2}(\sigma - q_{\mathrm{RT}}) \right\|_{L^2(\Omega)} = \operatorname*{argmin}_{\substack{q_{\mathrm{RT}} \in p_{N,\mathrm{RT}} + Q(\mathcal{T}) \\ \mathrm{div}\, q_{\mathrm{RT}} = -f_{\mathcal{T}}}} \left\| \mathbb{S}^{-1/2}(\sigma_h - q_{\mathrm{RT}}) \right\|_{L^2(\Omega)}.$$

Proof. The minimisation problem is characterised by the Lagrange functional

$$L(q_{\mathrm{RT}}, v_0) = \frac{1}{2} \left\| \mathbb{S}^{-1/2}(q_{\mathrm{RT}} - \sigma) \right\|^2_{L^2(\Omega)} + \int_\Omega v_0 (\mathrm{div}\, f_{\mathcal{T}} + \mathrm{div}\, q_{\mathrm{RT}}) \,\mathrm{d}x$$
$$\text{for } q_{\mathrm{RT}} \in \mathrm{RT}_0(\mathcal{T}) \text{ and } v_0 \in P_0(\mathcal{T}).$$

The optimality conditions for a minimiser (p_{RT}, u_0) of L read

$$\begin{aligned} \int_\Omega \mathbb{S}^{-1}(p_{\mathrm{RT}} - \sigma) \cdot q_{\mathrm{RT}} \,\mathrm{d}x + \int_\Omega \mathrm{div}\, q_{\mathrm{RT}} u_0 \,\mathrm{d}x = 0 \qquad & \text{for all } q_{\mathrm{RT}} \in Q(\mathcal{T}), \\ \int_\Omega \mathrm{div}\, p_{\mathrm{RT}} v_0 \,\mathrm{d}x = \int_\Omega f v_0 \,\mathrm{d}s \qquad & \text{for all } v_0 \in \mathcal{P}_0(\mathcal{T}). \end{aligned} \tag{2.9}$$

Input: initial triangulation $\mathcal{T}_0$, tolerance $\delta > 0$
for $\ell = 0, 1, 2, \ldots$ **do**
  SOLVE: *Compute discrete minimiser* $u_\ell \in V_\ell$;
  ESTIMATE: *Compute error estimator* η_ℓ;
  if $\eta_\ell \leqslant \delta$ **then return**;
  MARK:
    Compute refinement indicators $\eta(E)$ for every $E \in \mathcal{E}^\ell$ and mark sides $\mathcal{E}_M^\ell \subseteq \mathcal{E}^\ell$ for bisection
  REFINE: *Apply red-green-blue-refinement (Algorithm 2.2) with input* $\mathcal{T}_\ell$ *and* $\mathcal{E}_M^\ell$ *to get regular refinement* $\mathcal{T}_{\ell+1}$;

Algorithm 2.1: General layout of the AFEM loop.

An integration by parts and $\operatorname{div} q_{\mathrm{RT}} \in \mathcal{P}_0(\mathcal{T})$ shows

$$-\int_\Omega \mathbb{S}^{-1}\sigma \cdot q_{\mathrm{RT}} \,\mathrm{d}x = -\int_\Omega \nabla u \cdot q_{\mathrm{RT}} \,\mathrm{d}x = \int_\Omega u \operatorname{div} q_{\mathrm{RT}} \,\mathrm{d}x = \int_\Omega u_{\mathcal{T}} \operatorname{div} q_{\mathrm{RT}} \,\mathrm{d}x$$

where $u_{\mathcal{T}}$ is the piecewise integral mean of u, i.e., $u_{\mathcal{T}}|_T \coloneqq \fint_T u \,\mathrm{d}x$ for $T \in \mathcal{T}$. This leads to the equivalence of the solutions (p_{RT}, u_0) of (2.9) and $(p_{\mathrm{RT}}, u_0 + u_{\mathcal{T}})$ of (2.8). The proof of the second assertion is very similar and can be found in Braess (2007, Lemma 9.1 on page 181). □

Another result from Bahriawati and Carstensen (2005, Theorem 7.1) shows that Raviart-Thomas MFEM is in fact a postprocessing of the nonconforming Crouzeix-Raviart FEM in the sense that

$$p_{\mathrm{RT}} = \mathbb{S}\nabla_{\mathrm{NC}}\widehat{u}_{\mathrm{CR}} - f_{\mathcal{T}}(\bullet - \operatorname{mid}(\mathcal{T}))/n$$

with the Crouzeix-Raviart solution $\widehat{u}_{\mathrm{CR}}$ for the piecewise integral means $f_{\mathcal{T}}|_T \coloneqq \fint_T f \,\mathrm{d}x$ for $T \in \mathcal{T}$ and $g_{\mathcal{E}}|_E \coloneqq \fint_E g \,\mathrm{d}s$ for $E \in \mathcal{E}(\Gamma_N)$ instead of f and g. The function $\operatorname{mid}(\mathcal{T})$ is the piecewise element center, i.e. $\operatorname{mid}(\mathcal{T})|_T \coloneqq \operatorname{mid}(T)$ for $T \in \mathcal{T}$.

2.3.4 AFEM Algorithm

Algorithm 2.1 displays the principal loop of the adaptive finite element method (AFEM). The input is an initial triangulation $\mathcal{T}_0$ for the level $\ell = 0$. On every level ℓ the first step is to solve the discrete problem. Then, an error estimator measures the error (in this thesis in the energy norm) between the unknown exact solution u and the discrete solution u_ℓ. Local refinement indicators (that might be derived from local contributions of the error estimator) lead to a set of sides that are marked for refinement. This is explained in the subsequent Subsubsection 2.3.4.3. After the refinement the loop repeats until the error is below some given tolerance.

2.3.4.1 Remarks on Error Estimation

The optimality of the AFEM algorithm depends on the error estimator η_ℓ and on the refinement indicators. There are three important requirements.

(a) The error estimator should be *reliable*, i.e., $|||u - u_\ell||| \leqslant C_{\text{rel}}\eta_\ell$ for some reliability constant C_{rel}. *Guaranteed upper bounds* satisfy $C_{\text{rel}} = 1$ and are desirable to guarantee that the error is really below the target tolerance.
(b) The error estimator should be *efficient*, i.e., $\eta_\ell \leqslant C_{\text{eff}} |||u - u_\ell|||$ for some efficiency constant C_{eff}. This constant indicates the amount of overestimation and a constant close to 1 is desirable.
(c) The refinement indicators $\eta(E)$ for $E \in \mathcal{E}$ (that are not necessarily derived from the error estimator) should be reliable and efficient locally, i.e. $\eta(E)$ is equivalent to the energy error in a neighbourhood of E, to guarantee reasonable mesh refinement.

The explicit residual-based error estimator of Section 3.4 usually yields satisfactory and computational cheap refinement indicators whereas its huge overestimation makes it less useful as a termination criterion. More elaborate reliable, efficient and also sharp error estimators are introduced in Chapter 3.

2.3.4.2 Remarks on Marking

There are various marking strategies. The implementation for this thesis relies on *Dörfler marking*. For given refinement indicators $\eta(T)$ for each triangle $T \in \mathcal{T}$ and some constant $0 < \theta \leqslant 1$, the Dörfler marking selects a subset $\mathcal{T}_M \subseteq \mathcal{T}$ of minimal cardinality, such that

$$\sum_{T\in\mathcal{T}_M} \eta(T)^2 \geqslant \theta \sum_{T\in\mathcal{T}} \eta(T)^2.$$

Then, the set of marked sides reads $\mathcal{E}_M := \bigcup_{T\in\mathcal{T}_M} \mathcal{E}(T)$. For $\theta = 1$ this yields $\mathcal{T}_M = \mathcal{T}$ and therefore uniform mesh refinement. Unless indicated otherwise, $\theta = 0.5$ is the default value for all numerical experiments in this thesis. Similarly, refinement indicators for sides $\eta(E)$ for each $E \in \mathcal{E}$ can be applied and the Dörfler marking directly selects a set $\mathcal{E}_M \subseteq \mathcal{E}$ of minimal cardinality, such that

$$\sum_{E\in\mathcal{E}_M} \eta(E)^2 \geqslant \theta \sum_{E\in\mathcal{E}} \eta(E)^2.$$

2.3.4.3 Remarks on Mesh Refinement in 2D

The adaptive mesh refinement algorithm employs the red-green-blue refinement strategy. This involves the concept of the reference edge. For every triangle $T \in \mathcal{T}$ a reference edge $\text{ref}(T)$ is chosen, e.g. the longest edge of the triangle. This reference edge is bisected if any other edge of the triangle is marked for refinement.

Definition 2.3.5 (Red-Green-Blue-Refinement)**.** *Consider some triangle $T \in \mathcal{T}$ with $\partial T = E_1 \cup E_2 \cup E_3$ and reference edge E_1.*

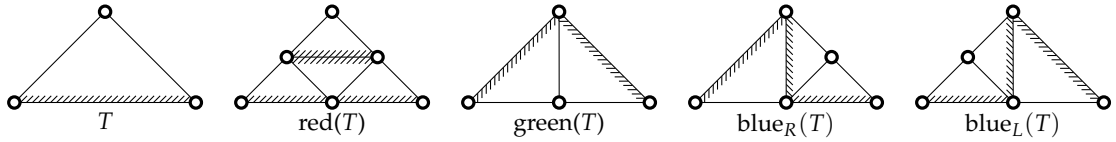

Figure 2.4: Unrefined triangle and its red-, green- and blue-refinements. The reference edges are hatched.

Input: triangulation $\mathcal{T}$, reference edges $\mathrm{ref}(T)$ for all $T \in \mathcal{T}$, marked edges $\mathcal{E}_M \neq \emptyset$
CLOSURE: *Extend $\mathcal{E}_M$ such that each $T \in \mathcal{T}$ with $\mathcal{E}(T) \cap \mathcal{E}_M \neq \emptyset$ satisfies* $\mathrm{ref}(T) \in \mathcal{E}_M$;
REFINE: *Refine each triangle with $\mathcal{E}(T) \cap \mathcal{E}_M \neq \emptyset$ separately by **red-refinement** (if all three edges are marked), **blue-refinement** (if two edges are marked) or **green-refinement** (if one edge is marked) such that each edge in $\mathcal{E}_M$ is bisected*;

Algorithm 2.2: Red-green-blue adaptive mesh refinement algorithm.

(a) The red-refinement $\mathrm{red}(T)$ *of T connects the midpoints of the edges E_1, E_2 and E_3 and so divides T into four congruent subtriangles.*
(b) The blue-refinement of T connects the midpoint of E_1 with the opposite node and with the midpoint of one of the remaining two edges and so divides T into three subtriangles.
(c) The green-refinement of $T \in \mathcal{T}$ connects the midpoint of E_1 with the opposite node and so divides T into two subtriangles.

Figure 2.4 displays all refinements and the inheritance of the (hatched) reference edges.

A red-refinement of each triangle $T \in \mathcal{T}$ results in the red-refined triangulation $\mathrm{red}(\mathcal{T})$ which is again regular. Algorithm 2.2 combines red-, green- and blue-refinement and thereby allows local refinement of a mesh while other areas remain untouched. Carefully note that the *closure* step in Algorithm 2.2 extends the set $\mathcal{E}_M$ to prevent hanging nodes which would lead to non-regular triangulations as in Figure 2.3. This algorithm follows Carstensen (2004) and yields a shape regular series of triangulations in the sense of Definition 2.3.6.

Definition 2.3.6 (Shape Regularity)**.** *A series of triangulations $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}_0}$ is shape regular if the quotient of the diameter h_T and radius r_T of the insphere of every $T \in \bigcup_{\ell \in \mathbb{N}_0} \mathcal{T}_\ell$ is bounded, i.e.,*

$$\max_{\ell \in \mathbb{N}_0} \max_{T \in \mathcal{T}_\ell} h_T / r_T \leqslant C.$$

Various important estimates require shape regular series of triangulations, see Theorem 3.4.5 for an example.

Figure 2.5 shows a square domain with a marked triangle (all three edges are marked). Then, the closure step in Algorithm 2.2 marks further edges to guarantee a regular triangulation after application of the red-green-blue-refinement. The resulting triangulation is depicted in the very right picture of Figure 2.5.

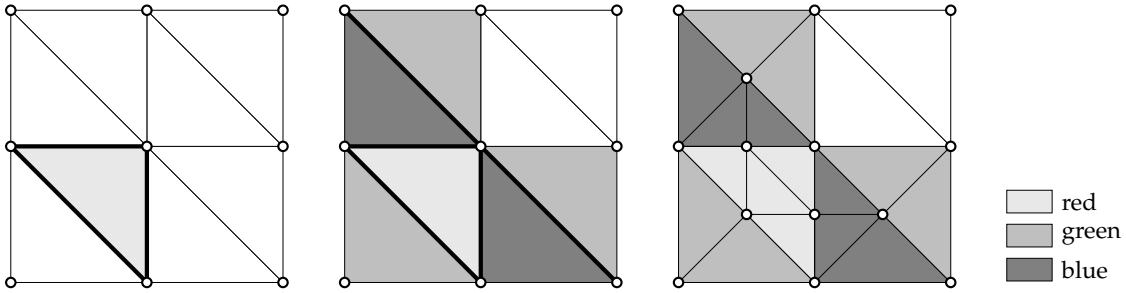

Figure 2.5: Exemplary adaptive mesh refinement of a square domain. The left picture shows the (thick) marked edges, the middle image shows the marked edges after closure and the right image shows the refined triangulation after red-green-blue refinement.

2.3.5 Implementation with the MATLAB Package AFEM

The implementation for this work was done in MATLAB and is based on the AFEM package by Carstensen and Numerical Analysis Group, HU Berlin (2009).

2.3.5.1 Basic Data Structures

Within this package any regular triangulation $\mathcal{T}$ in $n = 2$ dimensions is represented by two arrays named `c4n` and `n4e`. The array `c4n` of size $|\mathcal{N}| \times 2$ contains the two coordinates for the nodes $\mathcal{N}$ of the triangulation $\mathcal{T}$. The array `n4e` of size $|\mathcal{T}| \times 3$ contains the numbers of the three nodes of each element $T \in \mathcal{T}$ in counter-clockwise order. The row numbers in these arrays define the enumeration of the nodes and triangles. In the sequel the notation $T_k \in \mathcal{T}$ refers to the k-th triangle given by the k-th row `n4e(k,:)` of `n4e`. The first two nodes `n4e(k,1:2)` define the reference edge ref(T_k), which is additionally marked for refinement by the closure step in Algorithm 2.2 whenever another edge of the element is marked for refinement. The concept of the refinement edge prevents hanging nodes and ensures shape regularity, see Subsubsection 2.3.4.3 above.

The boundary edges of the triangulation are represented by pairs of node numbers and may be connected to Dirichlet or Neumann data. Depending on the type of the boundary data the node pairs are part of the arrays `n4sDb` or `n4sNb`. This completes the basic data structures that accompany any (user-defined) triangulation.

2.3.5.2 Side Enumerations

There are further enumerations derived from the basic data structures. The most important one is `n4s`, which is of size $|\mathcal{E}| \times 2$ and contains some enumeration of the edges in the triangulation. The k-th edge connects two nodes whose numbers are the entries of `n4s(k,:)`. Listing 2.1 of AFEM computes `n4s`. Line 2 collects all node pairings in `n4e` and line 3 and 4 remove the duplicates. The index set `ind` contains the element number of the first occurrence. That means that the ordering of the nodes prescribed in Line 4 is the same as in the first element that has this edge. This fixes an orientation of the normal vectors of the interior edges which is a very important issue for the implementation of the Raviart-Thomas finite element method.

```
function n4s=computeN4s(n4e)
allSides=[n4e(:,[1 2]); n4e(:,[2 3]); n4e(:,[3 1])];
[b,ind]=unique(sort(allSides,2),'rows','first');
n4s=allSides(sort(ind),:);
end
```

Listing 2.1: Listing for `computeN4s.m`

```
function s4n=computeS4n(n4e,n4s)
s4n = sparse(n4s(:,1),n4s(:,2),1:S,N,N);
s4n = s4n + s4n';
end
```

Listing 2.2: Listing for `computeS4n.m`

Another enumeration array `s4e` with size $|\mathcal{T}| \times 3$ contains the three edge numbers for each triangle in counter-clockwise order. For instance, `s4e(k,1)` is the number of the first edge between the first two nodes `n4e(k,[1 2])` of the k-th triangle and `s4e(k,3)` is the number of the third edge between the nodes `n4e(k,[3 1])`.

The structure `s4n` provides numbers of sides that connect neighbouring nodes. Since not every two nodes share an edge in the triangulation, `s4n` is a sparse matrix of size $|\mathcal{E}| \times |\mathcal{E}|$ with the entries `s4n(n4s(k,1),n4s(k,2))=k` for $k = 1,\ldots,|\mathcal{E}|$. Line 2 of Listing 2.2 represents this relation. Line 3 adds the transpose such that also the other permutation of the two nodes leads to the same result.

This gives only a little insight in the enumeration routines of `AFEM`. More enumerations will be discussed at some later point as needed.

2.3.5.3 Quadrature in `AFEM`

For elementwise integration on a triangulation given by `c4n` and `n4e` and `n4s`, the `AFEM` package offers the function `integrate`. This function computes Gauss quadrature points on squares and transforms them to quadrature points for each triangle with the conical product rule by Stroud (1971). The declaration of `integrate` reads

```
val = integrate(c4n,n4p,integrand,degree).
```

The second parameter is either `n4e` (for integration over elements) or `n4s` (for integration over sides). The fourth input parameter defines the accuracy of the quadrature. Polynomials up to `degree` are integrated exactly. The third input parameter defines the integrand function handle, which must satisfy the prototype

```
integrandval = integrand(n4p,Gpts4p,Gpts4ref).
```

The function `integrate` calls `integrand` inside a loop over all Gauss points. Each call transmits the input parameters `G4pts4p`, which is a list of the absolute coordinates of the current Gauss point in each part, and `Gpts4ref`, which contains the barycentric coordinates of the current Gauss point. The return value `integrandval` has to be of dimension `[nrParts n m]` where `nrParts` is the number of parts and `n` and `m` is the size of the function to be integrated, e.g. `n=m=1` for scalar-valued functions. The return value of `integrate` has the same dimensions.

As an example, the linear function $f(x_1, x_2) = x_1 + x_2$ is represented by the function handle `f=@(x)(x(:,1)+y(:,2))` and is exactly integrated (up to round-off errors) with

```
integrate(c4n,n4e,@(n4p,Gpts4p,Gpts4ref)(f(Gpts4p)),1).
```

2.3.5.4 Implementation of $\mathcal{P}_1$, $\mathcal{P}_2$ and CR Finite Elements

The computations of best-approximations with $\mathcal{P}_1$ or $\mathcal{P}_2$ finite elements involve local and global stiffness matrices and mass matrices for the discretisation of certain bilinear forms. For instance, the computation of the $\mathcal{P}_1$ best-approximation in the energy norm $|||\cdot||| := \|\nabla\cdot\|_{L^2(\Omega)}$ via the bilinear form a from Section 2.3.1 for $\mathbb{S} = \mathbb{I}$ leads to the stiffness matrix

$$A^{\mathcal{P}_1} := \left(\int_\Omega \nabla\varphi_k \cdot \nabla\varphi_j \,\mathrm{d}x\right)_{j,k=1,\dots,N}$$

for the $N = |\mathcal{N}|$ nodal basis functions $\{\varphi_1, \dots, \varphi_N\}$ of $\mathcal{P}_1(\mathcal{T}) \cap C(\Omega)$. The global stiffness matrix is a composition of local stiffness matrices. For the nodal basis functions $\varphi_{P_1}, \dots, \varphi_{P_{n+1}}$ on some element $T = \mathrm{conv}\{P_1, \dots, P_{n+1}\} \in \mathcal{T}$, the local stiffness matrix reads

$$A_T^{\mathcal{P}_1} := \left(\int_T \nabla\varphi_{P_j} \cdot \nabla\varphi_{P_k} \,\mathrm{d}x\right)_{j,k=1,\dots,n+1}$$

and can be computed by the following lemma.

Lemma 2.3.7. *On an element $T = \mathrm{conv}\{P_1, \dots, P_{n+1}\}$, it holds*

$$\begin{pmatrix} \nabla\varphi_{P_1}|_T \\ \vdots \\ \nabla\varphi_{P_{n+1}}|_T \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} 1 & \dots & 1 \\ P_1^{\mathrm{T}} & \dots & P_{n+1}^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \mathbb{I} \end{pmatrix} \in \mathbb{R}^{(n+1)\times n}.$$

Note that the P_j are row vectors and recall that the $\nabla\varphi_j$ are treated as column vectors.

Proof. The proof follows Lemma 1.23 from Carstensen (2009) and considers the representation

$$\varphi_{P_j}(x) = \frac{1}{n!\,|T|} \begin{vmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ P_1^{\mathrm{T}} & \dots & P_{j-1}^{\mathrm{T}} & x & P_{j+1}^{\mathrm{T}} & \dots & P_{n+1}^{\mathrm{T}} \end{vmatrix}$$

of the nodal basis function φ_{P_j} and the partial derivatives

$$\frac{\partial\varphi_{P_j}(x)}{\partial x_k} = \frac{1}{n!\,|T|} \begin{vmatrix} 1 & \dots & 1 & 0 & 1 & \dots & 1 \\ P_1^{\mathrm{T}} & \dots & P_{j-1}^{\mathrm{T}} & e_k & P_{j+1}^{\mathrm{T}} & \dots & P_{n+1}^{\mathrm{T}} \end{vmatrix} \quad \text{for } k = 1, \dots, n$$

where e_k is the k-th unit vector. Then, Cramer's rule for the linear system in the assertion

```
grads = [1 1 1;c4n(n4e(k,:),:)']\[0 0;eye(2)];
AlocalP1 = area * grads * grads';
AlocalP2(1:3,1:3) = AlocalP1.*(4*eye(3)-1);
Atmp = 4*ones(3,1)*AlocalP1([6 3 2]);
Atmp = Atmp-diag(diag(Atmp));
AlocalP2(1:3,[5 6 4]) = Atmp;
AlocalP2([5 6 4],1:3) = Atmp';
d = diag(AlocalP1);
d = d+d([2 3 1])+AlocalP1([2 6 3])';
Atmp2 = diag(d);
Atmp2([2 3 6]) = AlocalP1([3 6 2]);
Atmp2 = Atmp2 + Atmp2' - diag(diag(Atmp2));
AlocalP2(4:6,4:6) = 8*Atmp2;
AlocalP2 = AlocalP2/3;
```

Listing 2.3: Listing for the computation of $A_T^{\mathcal{P}_2}$ of the k-th triangle with the nodes `n4e(k,:)` in 2D.

and

$$\begin{vmatrix} 1 & \dots & 1 \\ P_1^{\mathrm{T}} & \dots & P_{n+1}^{\mathrm{T}} \end{vmatrix} = n! \, |T|$$

for the determinant of the system matrix concludes the proof. □

The `MATLAB` code line

```
grads=[1,1,1;c4n(n4e(k,:),:)']\[0,0;eye(2)]
```

exploits the formula of Lemma 2.3.7 to calculate the three gradients on the k-th triangle of the 2D triangulation given by `c4n` and `n4e`. Then, `grads*grads'` and a multiplication with the area $|T|$ of T gives the local stiffness matrix $A_T^{\mathcal{P}_1}$. The stiffness matrix of the $\mathcal{P}_2$ elements applies the characterisations in the proof of Theorem 2.2.5.(c) and the chain rule. With this, the gradients of the $\mathcal{P}_2$ basis functions read

$$\begin{aligned} \nabla \hat{\varphi}_{P_j} &= \nabla \left(\varphi_j (2\varphi_j - 1) \right) = (4\varphi_{P_j} - 1) \nabla \varphi_{P_j} \quad \text{for } j = 1, \dots, n+1 \quad \text{and} \\ \nabla \hat{\varphi}_{R_{jk}} &= \nabla \left(4\varphi_{P_j} \varphi_{P_k} \right) = 4 \left(\varphi_{P_k} \nabla \varphi_{P_j} + \varphi_{P_j} \nabla \varphi_{P_k} \right) \quad \text{for } 1 \leqslant j < k \leqslant n+1. \end{aligned}$$

Lengthy calculations and the application of Lemmas 2.2.18 and 2.3.7 to compute the products of nodal basis functions and their gradients result in the `MATLAB` Listing 2.3 for the computation of $A_T^{\mathcal{P}_2}$ in two dimensions.

The computation of the L^2 best-approximation $\operatorname{argmin}_{p \in \mathcal{P}_1(\mathcal{T})} \|p - f\|_{L^2(\Omega)}^2$ of some given function $f \in L^2(\Omega)$ leads to a linear system of equations $M^{\mathcal{P}_1} x = b$ that involves the mass matrix $M^{\mathcal{P}_k}$ with

$$M_{jk}^{\mathcal{P}_1} \coloneqq \int_\Omega \varphi_k \varphi_j \, \mathrm{d}x \quad \text{for } j, k = 1, \dots, N$$

and the nodal basis functions $\{\varphi_1, \dots, \varphi_N\}$ of $\mathcal{P}_1(\mathcal{T}) \cap C(\Omega)$.

Lemma 2.2.18 gives an explicit formula for the entries of the local $\mathcal{P}_1$ mass matrix on

some element $T \in \mathcal{T}$,

$$M_T^{\mathcal{P}_1} := \left(\int_T \varphi_j \varphi_k \, \mathrm{d}x \right)_{j,k=1,\dots,n+1} = \frac{|T| \, n!}{(n+2)!} (1 + \mathbb{I}).$$

Notice that $\mathbb{I}$ is the identity matrix and $1 + \mathbb{I}$ denotes, as in `MATLAB`, a $n \times n$ matrix with two on the diagonal and one in all other entries. The local mass matrix for the Crouzeix-Raviart basis functions of Theorem 2.2.6 leads to

$$\begin{aligned} M_T^{\mathrm{CR}} := \left(\int_T \psi_j \psi_k \, \mathrm{d}x \right)_{j,k=1,\dots,n+1} &= \left(\int_T (1 - n\varphi_j)(1 - n\varphi_k) \, \mathrm{d}x \right)_{j,k=1,\dots,n+1} \\ &= \frac{|T|}{(n+1)(n+2)} (2 - n + n^2 \mathbb{I}). \end{aligned}$$

Hence, M_T^{CR} is diagonal in $n = 2$ dimensions. The products of the $\mathcal{P}_2$ basis functions as in the proof of Theorem 2.2.5.(c) lead to products of up to four nodal basis functions that can also be computed by Lemma 2.2.18.

2.3.5.5 Implementation of Raviart-Thomas Elements

This section concludes with some remarks on the implementation of Raviart-Thomas elements. Theorem 2.2.7 already implies that a Raviart-Thomas function $q \in \mathrm{RT}(T)$ on a single element $T \in \mathcal{T}$ in two dimensions can be expressed by the Raviart-Thomas basis functions

$$\vartheta_j(x) := \frac{1}{2\,|T|} \left(x - P_j \right) \qquad \text{for } j = 1, \dots, 3. \tag{2.10}$$

These functions are identical to the ones in Bahriawati and Carstensen (2005) up to the factor $|E_j|$ which would have led to different properties in Theorem 2.2.7. In this thesis however, the functions ϑ_j are used to represent q and `getRTBasis` from Listing 2.4 allows the evaluation of the three basis function in the 2D case at any point $x \in T$. Up to the correct numbering of nodes and edges, the code is straightforward. Remember that P_1 corresponds to `n4e(:,1)` and E_1 from Figure 2.2 is the edge between P_2 and P_3. However, the outcome `val` of `getRTBasis` matches the numbering in the enumeration array `s4e` where the edge number of E_1 is at the second position. This leads to the index in Line 6 of Listing 2.4. Analogous considerations explain the indexing in Lines 7–8.

For the implementation of the mixed finite element method, which computes the L^2 best-approximation of the gradient, one also needs the mass matrix of the Raviart-Thomas basis functions. A neat formula is explained in Bahriawati and Carstensen (2005) and, up to slight modifications, transfers to the basis functions used in this thesis.

Lemma 2.3.8 (Bahriawati and Carstensen (2005))**.** *On any element* $T = \mathrm{conv}\{P_1, P_2, P_3\}$ *with the local Raviart-Thomas basis functions from* (2.10)*, it holds*

$$B_T = \left(\int_T \vartheta_j \cdot \vartheta_k \, \mathrm{d}x \right)_{j,k=1,\dots,3} = \frac{1}{48\,|T|} N^{\mathrm{T}} M N$$

```
function val=getRTBasis(pts,c4n,n4e,area4e)
% evaluates the three RT edge basis functions
areas=2*[area4e area4e];
nodes1=c4n(n4e(:,1),:);
nodes2=c4n(n4e(:,2),:);
nodes3=c4n(n4e(:,3),:);
val=zeros(size(n4e,1),2,3);
val(:,:,2)=(pts - nodes1)./areas;
val(:,:,3)=(pts - nodes2)./areas;
val(:,:,1)=(pts - nodes3)./areas;
end
```

Listing 2.4: Listing for `getRTBasis.m`

```
function B=LocalRTMassMatrix(c4n,n4e,area4e)
if nargin<3, area4e=computeArea4e(c4n,n4e); end
M=sparse([2 0 1 0 1 0 0 2 0 1 0 1 1 0 2 0 1 0 0 1 0 2 0 1 1 0 1 0 2 0 0 1 0 1 0 2]');
bigN=zeros(size(n4e,1),6,3);
p3p1=c4n(n4e(:,3),:)-c4n(n4e(:,1),:);
p3p2=c4n(n4e(:,3),:)-c4n(n4e(:,2),:);
p2p1=c4n(n4e(:,2),:)-c4n(n4e(:,1),:);
bigN(:,1:2,2)=p3p1;
bigN(:,1:2,3)=p3p2;
bigN(:,3:4,1)=-p3p1;
bigN(:,3:4,3)=-p2p1;
bigN(:,5:6,1)=-p3p2;
bigN(:,5:6,2)=p2p1;
bigM=permute(reshape(M*ones(1,size(n4e,1)),[6 6 size(n4e,1)]),[3 1 2]);
B=matMul(matMul(permute(bigN,[1 3 2]),matMul(bigM,bigN)),1./area4e)/48;
end
```

Listing 2.5: Listing for `LocalRTMassMatrix.m`

with

$$M = \begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{pmatrix} \in \mathbb{R}^{6\times 6} \quad \textit{and} \quad N = \begin{pmatrix} 0 & P_1 - P_2 & P_1 - P_3 \\ P_2 - P_1 & 0 & P_2 - P_3 \\ P_3 - P_1 & P_3 - P_2 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 3}.$$

The `MATLAB` function `LocalRTMassMatrix` of Listing 2.5 calculates the local mass matrices for the Raviart-Thomas basis functions by Lemma 2.3.8 for all triangles $T \in \mathcal{T}$ of some triangulation $\mathcal{T}$. The submatrices `bigM(j,:,:)` and `bigN(j,:,:)` contain the matrix M and N from Lemma 2.3.8 for the j-th triangle. The function `matMul` in Line 15 simultaneously multiplies two three-dimensional matrices elementwise along the first dimension and so `B(j,:,:)` contains the local mass matrix of the j-th triangle.

On the two triangles that share one side, the normal fluxes are the same up to the orientation of the normal vector they are based on. Therefore, to combine the local basis functions to global basis functions, one has to choose an orientation for the normal vector of each side. Let $E = \partial T_+ \cap \partial T_-$ be an interior edge which is shared by the two elements T_+ and T_- of a given triangulation. In this thesis we assume, as do Bahriawati and

```
function [sig4e,e4s] = computeSig4e(n4e)
s4e=computeS4e(n4e);
e4s=computeE4s(n4e);
e4s=sort(e4s,2);
ind=find(e4s(:,1)==0);
e4s(ind,1)=e4s(ind,2);
sig4e=-ones(size(n4e));
for k=1:3
    ind=find(e4s(s4e(:,k),2)==(1:size(s4e,1))');
    sig4e(ind,k)=1;
end
end
```

Listing 2.6: Listing for `computeSig4e.m`

Carstensen (2005), that T_- is listed *before* T_+ in the structure `n4e` and that ν_E is the outer unit normal vector of T_+ on E. Recall that the element number is given by the row number in `n4e`. The new common `AFEM` function `computeSig4e` of Listing 2.6 computes an array `sig4e` that contains a sign $+1$ or -1 for each of the three edges of every element. The structure matches `s4e` and its entries `sig4e(j,k)` $= (\nu_{E_k} \cdot \nu_{T_j})|_{E_j}$ define the sign of the normal vector of the k-th edge in the j-th triangle, i.e. `s4e(j,k)`. A sign 1 means that the normal vector of the edge has the same orientation as the outer normal vector of the triangle on that edge. A sign -1 means that the orientation is opposite.

3 Residual-Based Error Estimation

This chapter deals with basic ideas and novel aspects of unified error estimation in the spirit of the unified approach of Carstensen (2005) and Carstensen et al. (2012a).

3.1 Definitions and Motivation

Unified error estimation studies residuals of the form

$$\mathrm{Res}(v) \coloneqq \int_\Omega f v \,\mathrm{d}x + \int_{\Gamma_N} g v \,\mathrm{d}s - \int_\Omega \sigma_h \cdot \nabla v \,\mathrm{d}x \qquad \text{for } v \in H^1(\Omega) \tag{3.1}$$

for given data $f \in L^2(\Omega)$, $g \in L^2(\Gamma_N)$, some discrete flux $\sigma_h \in \mathcal{P}_0(\mathcal{T};\mathbb{R}^n)$ and a piecewise constant diffusion tensor $\mathbb{S} \in \mathcal{P}_0(\mathcal{T}_0;\mathbb{R}^{n\times n})$ with respect to some regular triangulation $\mathcal{T}_0$. The regular triangulation $\mathcal{T}$ is arbitrary but assumed to be a refinement of $\mathcal{T}_0$ to ensure $\mathbb{S} \in \mathcal{P}_0(\mathcal{T};\mathbb{R}^{n\times n})$. The smallest and largest eigenvalues of $\mathbb{S}$ on $T \in \mathcal{T}$ are denoted by

$$\lambda_{\min,T} \coloneqq \min(\mathrm{eig}(\mathbb{S}|_T)) = \lambda_{\min,\mathcal{T}}|_T \quad \text{and} \quad \lambda_{\max,T} \coloneqq \max(\mathrm{eig}(\mathbb{S}|_T)) = \lambda_{\max,\mathcal{T}}|_T. \tag{3.2}$$

They define the piecewise values of the functions $\lambda_{\min,\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$ and $\lambda_{\max,\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$. The smallest eigenvalue in Ω is denoted by $\lambda_{\min,\Omega}$.

Another notation of the residual (3.1) involves the normal jumps of σ_h.

Definition 3.1.1 (Normal Jumps of σ_h)**.** *The normal jump of σ_h on the side $E \in \mathcal{E}$ reads*

$$[\sigma_h \cdot \nu_E]_E \coloneqq \begin{cases} (\sigma_h|_{T_-} - \sigma_h|_{T_+}) \cdot \nu_{T_-} & \text{for } E = \partial T_- \cap \partial T_+ \in \mathcal{E}(\Omega), \\ \sigma_h \cdot \nu - g & \text{for } E \in \mathcal{E}(\Gamma_N), \\ 0 & \text{for } E \in \mathcal{E}(\Gamma_D). \end{cases}$$

An elementwise integration by parts transforms the residual (3.1) into

$$\mathrm{Res}(v) = \sum_{T\in\mathcal{T}} \int_T f v \,\mathrm{d}x - \int_{\partial T} \sigma_h \cdot \nu v \,\mathrm{d}s = \int_\Omega f v \,\mathrm{d}x - \sum_{E\in\mathcal{E}} \int_E [\sigma_h \cdot \nu]_E v \,\mathrm{d}s.$$

The dual norm of Res from Definition 2.1.3 with respect to some subspace $W \subset H^1(\Omega)$ and the energy norm $|||\cdot||| \coloneqq \left\|\mathbb{S}^{1/2}\nabla\cdot\right\|_{L^2(\Omega)}$ reads

$$|||\mathrm{Res}|||_{W^\star} \coloneqq \sup_{v\in W\setminus\{0\}} \mathrm{Res}(v)/\,|||v|||\,. \tag{3.3}$$

If $W = V = H^1_D(\Omega)$ the notation reduces to $|||\mathrm{Res}|||_\star$.

The end of this section recalls the Poisson model problem of Subsection 2.3.2 to motivate the relevance of these quantities. The $\mathcal{P}_1$ conforming finite element method leads to the residual

$$\mathrm{Res}(v) = a(u - u_h, v) = \int_\Omega fv \,\mathrm{d}x + \int_{\Gamma_N} gv \,\mathrm{d}s - \int_\Omega \mathbb{S}\nabla u_h \cdot \nabla v \,\mathrm{d}x \qquad \text{for } v \in V = H^1_D(\Omega)$$

and $|||\cdot||| = a(\cdot,\cdot)^{1/2}$. In case of homogeneous Dirichlet boundary conditions $u_D \equiv 0$, $a(u - u_h, \cdot)$ has the Riesz representation $e \coloneqq u - u_h \in H^1_D(\Omega)$, and the following Lemma from Brenner and Carstensen (2004) associates $|||\mathrm{Res}|||_\star$ to the energy norm $|||e|||$.

Lemma 3.1.2 (Error Residual Identity)**.** *For any $v \in V$ and the Riesz representation $e \in V$ with $\mathrm{Res}(v) = a(e, v)$ and $|||\cdot||| = a(\cdot,\cdot)^{1/2}$, it holds*

$$\frac{|||e||| - \mathrm{Res}(v)}{|||e|||} = \frac{1}{2}\left|\left|\left| v - \frac{e}{|||e|||} \right|\right|\right|^2.$$

Proof. The proof follows from elementary algebra. □

Lemma 3.1.2 has two consequences. Firstly, any $v \in V$ yields $\mathrm{Res}(v) \leqslant |||e|||$. Secondly, $v = e/|||e|||$ leads to the identity $\mathrm{Res}(v) = |||e|||$. Hence, $|||e||| = |||\mathrm{Res}|||_\star$. However, the computation of $v = e/|||e|||$ is as expensive as the computation of u itself and so it is more reasonable to strive for bounds of $|||\mathrm{Res}|||_\star$. While any $v \in V$ leads to guaranteed lower bounds of $|||\mathrm{Res}|||_\star$, the remaining parts of this chapter are devoted to the more elaborate topic of guaranteed upper bounds.

3.2 Equilibration A Posteriori Error Estimators

This section deals with equilibration a posteriori error estimators that lead to guaranteed upper error bounds for $|||\mathrm{Res}|||_\star$ or error majorants in the sense of Repin (1999).

3.2.1 Introduction to Equilibration

As a point of departure, an integration by parts in (3.1) for any $q \in H(\mathrm{div}, \Omega)$ with $q \cdot \nu \in L^2(\Gamma_N)$ yields

$$\mathrm{Res}(v) = \int_\Omega (f + \mathrm{div}\, q)v \,\mathrm{d}x + \int_{\Gamma_N} (g - q \cdot \nu)v \,\mathrm{d}s + \int_\Omega (q - \sigma_h) \cdot \nabla v \,\mathrm{d}x \qquad \text{for } v \in V. \tag{3.4}$$

All integrals define linear maps in $V^\star$ and their dual norms read

$$\begin{aligned}
\|\!| f + \operatorname{div} q |\!|\!|_\star &\coloneqq \sup_{v\in V\setminus\{0\}} \int_\Omega (f + \operatorname{div} q) v \,\mathrm{d}x \,/\, |\!|\!| v |\!|\!| \,, \\
|\!|\!| \gamma_{\Gamma_N}(g - q\cdot\nu) |\!|\!|_\star &\coloneqq \sup_{v\in V\setminus\{0\}} \int_{\Gamma_N} (g - q\cdot\nu) v \,\mathrm{d}s \,/\, |\!|\!| v |\!|\!| \,, \\
|\!|\!| \operatorname{div}(\sigma - q) |\!|\!|_\star &\coloneqq \sup_{v\in V\setminus\{0\}} \int_\Omega (q - \sigma_h)\cdot\nabla v \,\mathrm{d}x \,/\, |\!|\!| v |\!|\!| \,.
\end{aligned} \tag{3.5}$$

The term $\gamma_{\Gamma_N}(g - q\cdot\nu)$ relates to the trace of $g - q\cdot\nu \in L^2(\Gamma_N)$ along Γ_N. The notation $\operatorname{div}(q - \sigma_h) \in V^\star$ relates to some pseudo divergence that exists only as a linear functional in $V^\star$ and maps $v \in V$ to

$$\operatorname{div}(q - \sigma_h)(v) \coloneqq -\int_\Omega (q - \sigma_h)\cdot\nabla v \,\mathrm{d}x = \int_\Omega \operatorname{div}_{\mathcal{T}}(q - \sigma_h) v \,\mathrm{d}x - \sum_{E\in\mathcal{E}} \int_E [\sigma_h\cdot\nu] v \,\mathrm{d}s.$$

The two latter terms stem from an elementwise integration by parts where $\operatorname{div}_{\mathcal{T}}$ is the piecewise divergence operator.

A split of all integrals in (3.4) into integrals over triangle domains for all $T \in \mathcal{T}$ leads to

$$\operatorname{Res}(v) = \sum_{T\in\mathcal{T}} \left(\frac{\int_\Omega (q - \sigma_h)\cdot\nabla v \,\mathrm{d}x}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} + \frac{\int_T (f + \operatorname{div} q) v \,\mathrm{d}x}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} \frac{\int_E (g - q\cdot\nu) v \,\mathrm{d}s}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} \right) \times \|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)} \,.$$

Then, the supremum over all $v \in V$, a Cauchy inequality in $\mathbb{R}^{|T|}$ and division by $|\!|\!| v |\!|\!|$ show

$$\begin{aligned}
|\!|\!| \operatorname{Res} |\!|\!|_\star^2 \leqslant \sup_{v\in V} \sum_{T\in\mathcal{T}} \Bigg(& \frac{\int_\Omega (q - \sigma_h)\cdot\nabla v \,\mathrm{d}x}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} + \frac{\int_T (f + \operatorname{div} q) v \,\mathrm{d}x}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} \\
& + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} \frac{\int_E (g - q\cdot\nu) v \,\mathrm{d}s}{\|\mathbb{S}^{1/2}\nabla v\|_{L^2(T)}} \Bigg)^2 .
\end{aligned} \tag{3.6}$$

Note that this allows slightly sharper estimates than those from Carstensen and Merdon (2013), which are based on the triangle inequality

$$|\!|\!| \operatorname{Res} |\!|\!|_\star \leqslant |\!|\!| \operatorname{div}(\sigma - q) |\!|\!|_\star + |\!|\!| f + \operatorname{div} q |\!|\!|_\star + |\!|\!| \gamma_{\Gamma_N}(g - q\cdot\nu) |\!|\!|_\star \,. \tag{3.7}$$

Equilibration is the design of some $q \in H(\operatorname{div}, \Omega)$ that (almost) satisfies the equilibration conditions $\operatorname{div} q + f = 0$ in Ω and $g - q\cdot\nu = 0$ along Γ_N such that the last two terms in (3.6) vanish (or are of higher order). For instance, designs with $\operatorname{div} + f_{\mathcal{T}} = 0$ in Ω and $g_{\mathcal{E}} - q\cdot\nu = 0$ along Γ_N, allow to bound the last two terms in (3.6) essentially by the

Figure 3.1: Example for some regular triangulation (left) and its dual mesh (right).

oscillations

$$\operatorname{osc}(f,\mathcal{T}) := \left(\sum_{T\in\mathcal{T}} \|h_T(f-f_T)\|^2_{L^2(T)}\right)^{1/2} = \|h_{\mathcal{T}}(f-f_{\mathcal{T}})\|_{L^2(\Omega)},$$
$$\operatorname{osc}(g,\mathcal{E}(\Gamma_N)) := \left(\sum_{E\in\mathcal{E}(\Gamma_N)} \left\|h_E^{1/2}(g-g_E)\right\|^2_{L^2(E)}\right)^{1/2} = \left\|h_{\mathcal{E}}^{1/2}(g-g_{\mathcal{E}})\right\|_{L^2(\Gamma_N)}.$$

Here $f_{\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$ and $g_{\mathcal{E}} \in \mathcal{P}_0(\mathcal{E}(\Gamma_N))$ are the piecewise integral means of f and g, i.e.,

$$f_{\mathcal{T}}|_T := \fint_T f\,\mathrm{d}x \text{ for } T\in\mathcal{T} \quad \text{and} \quad g_{\mathcal{E}}|_E := \fint_E g\,\mathrm{d}s \text{ for } E\in\mathcal{E}(\Gamma_N).$$

For (piecewise) H^1 functions $f \in H^1(\mathcal{T})$ or $g \in H^1(\mathcal{E})$, a piecewise Poincaré inequality reveals the higher-order property in the sense that

$$\operatorname{osc}(f,\mathcal{T}) \leqslant C_{\mathrm{P}}(\mathcal{T})\left\|h_{\mathcal{T}}^2\nabla f\right\|_{L^2(\Omega)} \text{ and } \operatorname{osc}(g,\mathcal{E}(\Gamma_N)) \leqslant C_{\mathrm{P}}(\mathcal{E}(\Gamma_N))\left\|h_{\mathcal{E}}^{3/2}\partial g/\partial s\right\|_{L^2(\Gamma_N)}$$

where $C_{\mathrm{P}}(\mathcal{T}) := \max_{T\in\mathcal{T}} C_{\mathrm{P}}(T)$ and $C_{\mathrm{P}}(\mathcal{E}(\Gamma_N)) := \max_{E\in\mathcal{E}(\Gamma_N)} C_{\mathrm{P}}(E)$ are the maximal Poincaré constants for the set of elements and Neumann sides, respectively.

The first term in (3.6) dominates and is often (suboptimally) estimated by a Hölder inequality in

$$\int_T (q-\sigma_h)\nabla v\,\mathrm{d}x \leqslant \left\|\mathbb{S}^{-1/2}(\sigma-q)\right\|_{L^2(T)}\left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}.$$

Similarly, the first term in (3.7) is bounded by

$$|||\operatorname{div}(\sigma-q)|||_\star \leqslant \left\|\mathbb{S}^{-1/2}(\sigma-q)\right\|_{L^2(\Omega)}. \tag{3.8}$$

This term can be minimised by mixed or least-square finite element methods in some adequately subspace of $H(\operatorname{div},\Omega)$, e.g. Raviart-Thomas finite elements $\mathrm{RT}_0(\mathcal{T})$. However, this effort might be considered too expensive. The novel equilibration error estimators in Subsections 3.2.2 and 3.2.3 provide local designs of suitable $q \in H(\operatorname{div},\Omega)$ and explicit upper bounds for all terms in (3.6), thence lead to fully computable guaranteed upper bounds for $|||\mathrm{Res}|||_\star$. Section 3.3 proposes a postprocessing for any equilibration error estimator and so leads to an improved estimate of (3.8).

```
function [c4n,n4e_fine,n4sDb,n4sNb,parents4e]=refineDual(c4n,n4e,n4sDb,n4sNb)
n4s=computeN4s(n4e);
s4e=computeS4e(n4e);
s4n=computeS4n(n4e);
midpoint4e=computeMid4e(c4n,n4e);
midpoint4s=computeMid4s(c4n,n4s);
nrElems=size(n4e,1);
nrNodes=size(c4n,1);
nrNbEdges= size(n4sNb,1);
c4n=[c4n;midpoint4e;midpoint4s];
n4e_fine=zeros(nrElems*6,3);
n4e_fine(1:6:end,:)=[n4e(:,1) nrNodes+[1:nrElems]' nrNodes+nrElems+s4e(:,3)];
n4e_fine(2:6:end,:)=[n4e(:,1) nrNodes+nrElems+s4e(:,1) nrNodes+[1:nrElems]'];
n4e_fine(3:6:end,:)=[n4e(:,2) nrNodes+[1:nrElems]' nrNodes+nrElems+s4e(:,1)];
n4e_fine(4:6:end,:)=[n4e(:,2) nrNodes+nrElems+s4e(:,2) nrNodes+[1:nrElems]'];
n4e_fine(5:6:end,:)=[n4e(:,3) nrNodes+[1:nrElems]' nrNodes+nrElems+s4e(:,2)];
n4e_fine(6:6:end,:)=[n4e(:,3) nrNodes+nrElems+s4e(:,3) nrNodes+[1:nrElems]'];
parents4e=reshape(ones(6,1)*(1:nrElems),[1 nrElems*6]);
n4sDb=refineBoundary(n4sDb,s4n,nrNodes+nrElems);
n4sNb=refineBoundary(n4sNb,s4n,nrNodes+nrElems);
end

function n4sB=refineBoundary(n4sB,s4n,offset)
if isempty(n4sB), return; end
midB=s4n(n4sB(:,1),n4sB(:,2));
midB=full(diag(midB));
midB=reshape(midB+offset,1,[]);
n4sB=n4sB';
n4sB=[n4sB(1,:);midB;midB;n4sB(2,:)];
n4sB=reshape(n4sB,2,[])';
end
```

Listing 3.1: Listing for `refineDual.m`

3.2.2 Design by Luce-Wohlmuth

Luce and Wohlmuth (2004) solve local problems around each node on the dual triangulation $\mathcal{T}^\star$ of $\mathcal{T}$ with sides $\mathcal{E}^\star$ and nodes $\mathcal{N}^\star$ and compute some $q_{\mathrm{LW}} \in \mathrm{RT}_0(\mathcal{T}^\star)$.

The dual triangulation $\mathcal{T}^\star$ is well-known in the finite volume methodology. In $n = 2$ dimensions, it connects each center $\mathrm{mid}(T)$ of an element $T \in \mathcal{T}$ with the side midpoints $\mathrm{mid}(\mathcal{E}(T))$ and nodes $\mathcal{N}(T)$ and thus divides each element $T \in \mathcal{T}$ into six subelements of area $|T|/6$ and every side $E \in \mathcal{E}$ into two subsides of length $|E|/2$. Figure 3.1 shows some example triangulation and its dual mesh. Listing 3.1 displays MATLAB code that generates the refined data structures `c4n_fine`, `n4e_fine`, `n4sDb_fine` and `n4sNb_fine` for the dual mesh from the given triangulation data structures `n4e`, `c4n`, `n4sDb` and `n4sNb`. For this, Line 10 enriches the nodes `c4n` by the coordinates of the element centers and the edge midpoints. Every block `n4e_fine(6*(j-1):6*j,:)` generated in Lines 12–17 consists of the six child elements of the j-th triangle `n4e(j,:)`. The array `parents4e` contains the parent number in `n4e` for each child in `n4e_fine`. The rest of the code generates the refined boundary data by bisection.

The dual mesh concept extends to tetrahedra T in $n = 3$ dimensions. Here also the edge midpoints are connected with the nodes and sides of T. This leads to 24 subtetrahedra of equal volume and every face $E \in \mathcal{E}(T)$ is divided into six triangles of area $|E|/6$.

For some vertex $z \in \mathcal{N}$, $\varphi_z^\star$ denotes its nodal basis function with respect to the fine

BC type	$\dim Q(\mathcal{T}^\star(z))$
interior node (no BC)	1
Dirichlet/Dirichlet	1
Dirichlet/Neumann	0
Neumann/Neumann	0

Table 3.1: Dimension of $Q(\mathcal{T}^\star(z))$ for different boundary conditions (BC) along $\partial\omega_z^\star \cap \partial\Omega$ for $n = 2$ dimensions.

patch $\omega_z^\star := \{\varphi_z^\star > 0\}$ of the dual mesh $\mathcal{T}^\star$ and the neighbouring elements $\mathcal{T}^\star(z) := \{T^\star \in \mathcal{T}^\star \,|\, z \in \mathcal{N}^\star(T)\}$. Since $\sigma_h \in \mathcal{P}_0(\mathcal{T};\mathbb{R}^n)$ is continuous along $\partial\omega_z^\star \cap T$ for any $T \in \mathcal{T}$, the condition $q \cdot \nu = \sigma_h \cdot \nu \in \mathcal{P}_0(\mathcal{E}^\star(\partial\omega_z^\star))$ yields well-defined Neumann data along the exterior boundary edges $\mathcal{E}^\star(\partial\omega_z^\star)$ of $\omega_z^\star$. The solve of the following local problems determines the remaining fluxes over the interior edges $\mathcal{E}^\star(z) := \{E \in \mathcal{E}^\star \,|\, z \in E\}$.

The suggested design employs an interpolation $f^\star \in \mathcal{P}_0(\mathcal{T}^\star)$ of $f \in L^2(\Omega)$ defined by

$$f^\star|_{T^\star} := (n+1) \fint_T f\varphi_z \,\mathrm{d}x \quad \text{for the } n! \text{ subelements } T^\star \in \mathcal{T}^\star \text{ with } \mathcal{N}^\star(T^\star) \cap \mathcal{N}(T) = \{z\}. \tag{3.9}$$

Similarly, in case of Neumann data $g \in L^2(\Gamma_N)$, define $g^\star \in \mathcal{P}_0(\mathcal{E}^\star(\Gamma_N))$ by

$$g^\star|_{E^\star} := n \fint_E g\varphi_z \,\mathrm{d}x \quad \text{for the } (n-1)! \text{ subsides } E^\star \in \mathcal{E}^\star \text{ with } \mathcal{N}^\star(E^\star) \cap \mathcal{N}(E) = \{z\}. \tag{3.10}$$

This suffices to set up the local spaces

$$Q(\mathcal{T}^\star(z)) := \big\{\tau_h \in \mathrm{RT}_0(\mathcal{T}^\star(z)) \,\big|\, \operatorname{div}\tau_h + f^\star = 0 \text{ in } \omega_z^\star \;\&\; \tau_h\cdot\nu = \sigma_h\cdot\nu \text{ along } \partial\omega_z^\star\backslash\partial\Omega \;\&\; \tau_h\cdot\nu = g^\star \text{ along } \partial\omega_z^\star \cap \Gamma_N\big\}$$

and the mixed finite element method on $\mathcal{T}^\star(z)$ (see Lemma 2.3.4) computes

$$q_{\mathrm{LW}}|_{\omega_z^\star} := \operatorname*{argmin}_{\tau_h \in Q(\mathcal{T}^\star(z))} \left\| \mathbb{S}^{-1/2}(q_h - \tau_h) \right\|_{L^2(\omega_z^\star)}. \tag{3.11}$$

Alternatively, Luce and Wohlmuth (2004) and Braess and Schöberl (2008) describe an explicit design for the 2D case that influenced the implementation for this thesis. The design here is very close to recent suggestions by Vohralík (2011) for finite volume methods and mainly differs in the choice of $f^\star$. The next Lemma proves solvability of the local problems.

Lemma 3.2.1. *If $z \in \mathcal{M}$ and $\mathrm{Res}(\varphi_z) = 0$, the affine space $Q(\mathcal{T}^\star(z))$ is non-empty. If $z \in \mathcal{N}(\Gamma_D)$, the space $Q(\mathcal{T}^\star(z))$ is non-empty (without further assumptions). For $n = 2$ dimensions, the space $Q(\mathcal{T}^\star(z))$ has dimension 1 or 0 depending on the boundary case as given in Table 3.1.*

Proof. The proof consists of three steps.

Proof of non-emptiness of $Q(\mathcal{T}^\star(z))$. The search of some $q \in Q(\mathcal{T}^\star(z))$ for a free node $z \in \mathcal{M}$ describes a Neumann problem and requires the equilibration condition of the constraints, namely,

$$\int_{\partial\omega_z^\star \cap \Gamma_N} g^\star \,\mathrm{d}s + \int_{\partial\omega_z^\star \setminus \Gamma_N} \sigma_h \cdot \nu \,\mathrm{d}s = -\int_{\omega_z} f\varphi_z \,\mathrm{d}x. \tag{3.12}$$

An integration by parts, $\operatorname{div}\sigma_h = 0$ on every $T \in \mathcal{T}^\star$, and $\int_E \varphi_z \,\mathrm{d}s = |E|/n$, yield

$$\begin{aligned}
&\int_{\partial\omega_z^\star \cap \Gamma_N} g^\star \,\mathrm{d}s + \int_{\partial\omega_z^\star \setminus \Gamma_N} \sigma_h \cdot \nu \,\mathrm{d}s \\
&\qquad = \int_{\partial\omega_z^\star \cap \Gamma_N} g^\star - \sigma_h \cdot \nu \,\mathrm{d}s + \sum_{T \in \mathcal{T}^\star(z)} \int_{\partial T} \sigma_h \cdot \nu \,\mathrm{d}s - \sum_{E \in \mathcal{E}^\star(z)\setminus(\mathcal{E}^\star(\Gamma_N))} \int_E [\sigma_h \cdot \nu_E]_E \,\mathrm{d}s \\
&\qquad = \int_{\partial\omega_z \cap \Gamma_N} \varphi_z (g - \sigma_h \cdot \nu) \,\mathrm{d}s - \sum_{E \in \mathcal{E}(z)\setminus(\mathcal{E}(\Gamma_N))} \int_E \varphi_z [\sigma_h \cdot \nu_E]_E \,\mathrm{d}s \\
&\qquad = \int_{\Gamma_N} g\varphi_z \,\mathrm{d}s - \int_\Omega \sigma_h \cdot \nabla\varphi_z \,\mathrm{d}x.
\end{aligned}$$

Hence, (3.12) is equivalent to $\operatorname{Res}(\varphi_z) = 0$. For Dirichlet boundary nodes $z \in \mathcal{N}(\Gamma_D)$, the boundary $\partial\omega_z^\star \cap \partial\Omega$ has at least one unconstrained Dirichlet side and the equilibration condition (3.12) is not necessary. This concludes the proof for the existence of some solution $q \in Q(\mathcal{T}^\star(z))$.

Proof of $\dim Q(\mathcal{T}^\star(z)) = 1$ *for interior nodes* $z \in \mathcal{N}(\Omega)$. The complete set of solutions reads $q + Q_0(\mathcal{T}^\star(z))$ with any $q \in Q(\mathcal{T}^\star(z))$ and

$$\begin{aligned}
Q_0(\mathcal{T}^\star(z)) \coloneqq \big\{ \tau_h \in \mathrm{RT}_0(\mathcal{T}^\star(z)) \,\big|\, &\operatorname{div}\tau_h = 0 \text{ in } \omega_z^\star \;\&\; \tau_h \cdot \nu = 0 \text{ along } \partial\omega_z^\star \setminus \Gamma_D \\
&\;\&\; \tau_h \cdot \nu = 0 \text{ along } \partial\omega_z^\star \cap \Gamma_N \big\}.
\end{aligned}$$

The dimension of $Q(\mathcal{T}^\star(z))$ equals the dimension of $Q_0(\mathcal{T}^\star(z))$. Consider some element $q_0 \in Q_0(\mathcal{T}^\star(z))$. Since $\operatorname{div} q_0 = 0$, it follows $q_0 \in \mathcal{P}_0(\mathcal{T}; \mathbb{R}^2)$ and the discrete Helmholtz decomposition (Theorem 2.2.20) yields

$$q_0 = \nabla_{\mathrm{NC}}\alpha + \operatorname{Curl}\beta$$

for some $\alpha \in \mathrm{CR}_0(\mathcal{T}^\star(z))$ and $\beta \in \mathcal{P}_1(\mathcal{T}^\star(z)) \cap C(\omega_z^\star)$ with $\int_{\omega_z^\star} \beta \,\mathrm{d}x = 0$. An elementwise integration by parts and $\operatorname{div}\operatorname{Curl}\beta = 0$ shows

$$\int_{\omega_z^\star} \nabla_{\mathrm{NC}}\alpha \cdot \nabla_{\mathrm{NC}}\alpha \,\mathrm{d}x = \int_{\omega_z^\star} \operatorname{div} q_0\alpha \,\mathrm{d}x - \sum_{E \in \mathcal{E}^\star(z)} \int_E \alpha [q_0 \cdot \nu_E] \,\mathrm{d}s - \int_{\partial\omega_z^\star} \alpha q_0 \cdot \nu \,\mathrm{d}s = 0.$$

Hence, $\alpha \in \mathcal{P}_0(\mathcal{T}; \mathbb{R}^2)$ and, because of the zero boundary on α, it follows $\alpha \equiv 0$. It remains $q_0 = \operatorname{Curl}\beta$. The condition $q_0 \cdot \nu = \nabla\beta \cdot \tau = 0$ along $\partial\omega_z$ leads to $\beta = c\varphi_z^\star$ for some constant $c \in \mathbb{R}$. This proves $\dim Q(\mathcal{T}^\star(z)) = 1$.

Proof of $\dim Q(\mathcal{T}^\star(z)) \in \{0, 1\}$ *for boundary nodes* $z \in \mathcal{N}(\partial\Omega)$. For boundary nodes $z \in \mathcal{N}(\partial\Omega)$, a similar argumentation shows $q_0 = \operatorname{Curl}\beta$ for some $\beta \in \mathcal{P}_1(\mathcal{T}^\star(z)) \cap C(\omega_z^\star)$

n	$C_{\mathrm{P}}(T)$	$C_{\mathrm{N}}(E)$	C_n
1	-	-	3.3094
2	0.2610	1.1474	4.6742
3	0.3183	1.9397	6.0596

Table 3.2: Values for the constants $C_{\mathrm{P}}(T)$, $C_{\mathrm{N}}(E)$ and C_n from Theorem 3.2.2 for triangulations into elements that are shape-similar to T_{ref} (see Remark 3.2.3).

with $\int_{\omega_z^\star} \beta \,\mathrm{d}x = 0$ and $q_0 \cdot \nu = \nabla\beta \cdot \tau = 0$ along $\partial\omega_z \backslash \partial\Omega$. The remaining boundary $\partial\omega_z^\star \cap \partial\Omega = E_1 \cap E_2$ consists of two sides E_1 and E_2 with

$$0 = \int_{\omega_z^\star} \operatorname{div} q_0 \,\mathrm{d}x = \int_{\partial\omega_z^\star} q_0 \cdot \nu \,\mathrm{d}s = \int_{E_1\cap E_2} \nabla\beta \cdot \tau \,\mathrm{d}s.$$

This means that the flux on E_1 is the exact opposite of the flux on E_2. In case of one constrained Neumann side both fluxes are fixed (by zero) and it follows $\beta \in \mathcal{P}_0(\omega_z^\star)$, hence $\beta = 0$ and $\dim Q_0(\mathcal{T}^\star(z)) = 0$. Otherwise, in case of full Dirichlet boundary, we have again $\beta = c\varphi_z^\star$ and $\dim Q_0(\mathcal{T}^\star(z)) = 1$. This leads to the dimensions shown in Table 3.1 and concludes the proof. □

The following theorem states the equilibration properties and explicit constants for this design.

Theorem 3.2.2. *The constants* $C_n := 1 + (n+1)^2\,(2n!/(2+n)!)^{1/2}$ *and*

$$C_{\mathrm{N}}(E) := \left(\frac{\operatorname{diam}(\omega_E)^2}{|\omega_E|}\left(C_{\mathrm{P}}(\omega_E)^2 + 2C_{\mathrm{P}}(\omega_E)/n\right)\right)^{1/2} \quad \text{for } E \in \mathcal{E}(\Gamma_N)$$

and $q = q_{\mathrm{LW}}$ *bound the terms in* (3.6)*, for all* $T \in \mathcal{T}$ *and all* $E \in \mathcal{E}(\Gamma_N) \cap \mathcal{E}(T)$*, by*

$$\begin{aligned}
\frac{\int_T (f - f^\star) v \,\mathrm{d}x}{\left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}} &\leqslant C_{\mathrm{P}}(T)\left\|h_T\lambda_{\min,T}^{-1/2}(f - f^\star)\right\|_{L^2(T)} \leqslant C_{\mathrm{P}}(T)C_n\left\|h_T\lambda_{\min,T}^{-1/2}(f - f_T)\right\|_{L^2(T)},\\
\frac{\int_E (g - g^\star) v \,\mathrm{d}x}{\left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}} &\leqslant C_{\mathrm{N}}(E)\left\|h_E^{1/2}\lambda_{\min,T}^{-1/2}(g - g^\star)\right\|_{L^2(T)}\\
&\leqslant C_{\mathrm{N}}(E)C_{n-1}\left\|h_E^{1/2}\lambda_{\min,T}^{-1/2}(g - g_E)\right\|_{L^2(E)}.
\end{aligned}$$

The associated dual norms from (3.5) *for* $q = q_{\mathrm{LW}}$ *are bounded by*

$$\begin{aligned}
|||f - f^\star|||_\star &\leqslant C_{\mathrm{P}}(\mathcal{T})\left\|h_{\mathcal{T}}\lambda_{\min,\mathcal{T}}^{-1/2}(f - f^\star)\right\|_{L^2(\Omega)} \leqslant C_{\mathrm{P}}(\mathcal{T})C_n \operatorname{osc}(f\lambda_{\min,\mathcal{T}}^{-1/2}, \mathcal{T}),\\
|||\gamma_{\Gamma_N}(g - g^\star)|||_\star &\leqslant C_{\mathrm{N}}(\mathcal{E}(\Gamma_N))\left\|h_{\mathcal{E}}^{1/2}\lambda_{\min,\mathcal{E}}^{-1/2}(g - g^\star)\right\|_{L^2(\Gamma_N)}\\
&\leqslant C_{\mathrm{N}}(\mathcal{E}(\Gamma_N))C_{n-1}\operatorname{osc}(g\lambda_{\min,\mathcal{T}}^{-1/2}, \mathcal{E}(\Gamma_N)).
\end{aligned}$$

Proof. Proof of assertions that involve $f^\star$. With $|T \cap \omega_z^\star| = |T|/(n+1)$ and the partition of

unity property, it follows

$$\int_T f^\star \,\mathrm{d}x = \sum_{z\in\mathcal{N}(T)} \int_{T\cap\omega_z^\star} f^\star \,\mathrm{d}x = \sum_{z\in\mathcal{N}(T)} \frac{(n+1)\,|T\cap\omega_z^\star|}{|T|} \int_T f\varphi_z \,\mathrm{d}x = \int_T f \,\mathrm{d}x.$$

This orthogonality, a Cauchy inequality and a Poincaré inequality on T with Poincaré constant $C_P(T)$ result in

$$\begin{aligned}\int_T (f-f^\star)v \,\mathrm{d}x &= \int_T (f-f^\star)(v-v_{\mathcal{T}}) \,\mathrm{d}x \\ &\leqslant C_P(T) \left\| h_T \lambda_{\min,T}^{-1/2} (f-f^\star) \right\|_{L^2(T)} \lambda_{\min,T}^{1/2} \|\nabla v\|_{L^2(T)}.\end{aligned}$$

A triangle inequality yields

$$\|f-f^\star\|_{L^2(T)} \leqslant \|f-f_T\|_{L^2(T)} + \left(\sum_{z\in\mathcal{N}(T)} \|f_T - f^\star\|^2_{L^2(T\cap\omega_z^\star)} \right)^{1/2}. \tag{3.13}$$

The formula $\|\varphi_z\|^2_{L^2(T)} = n!\,|T|\,2/(2+n)!$ from Lemma 2.2.18 and $|T\cap\omega_z^\star| = |T|/(n+1)$ show

$$\begin{aligned}\|f_T - f^\star\|^2_{L^2(T\cap\omega_z^\star)} &= (n+1)^2 \left(\int_T (f-f_T)\varphi_z \,\mathrm{d}x \right)^2 |T\cap\omega_z^\star| \,/\, |T|^2 \\ &= (n+1)^3 \,\|(f-f_T)\varphi_z\|^2_{L^1(T)} \,|T|^{-1} \\ &\leqslant (n+1)^3 \,\|\varphi_z\|^2_{L^2(T)} \,\|f-f_T\|^2_{L^2(T)} \,|T|^{-1} \\ &= (n+1)^3 (2n!/(2+n)!) \,\|f-f_T\|^2_{L^2(T)}.\end{aligned}$$

The summation over all $n+1$ nodes $z\in\mathcal{N}(T)$ reveals

$$\sum_{z\in\mathcal{N}(T)} \|f_T - f^\star\|^2_{L^2(T\cap\omega_z^\star)} \leqslant (n+1)^4 (2n!/(2+n)!) \;\|f-f_T\|^2_{L^2(T)}$$

and the combination with (3.13) concludes the proof of the first assertion. The global estimate for $|||f-f^\star|||_\star$ follows from a Cauchy inequality in $\mathbb{R}^{|\mathcal{T}|}$.

Proof of assertions that involve $g^\star$. Consider some Neumann boundary side $E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)$ and some test function $v\in V$. Since $\int_E (g-g^\star)\,\mathrm{d}s = 0$, one can subtract an arbitrary constant $v_E\in\mathcal{P}_0(E)$ such that

$$\int_E (g-g^\star)v \leqslant \|g-g^\star\|_{L^2(E)} \,\|v-v_E\|_{L^2(E)}.$$

The trace identity (Lemma 2.2.17) on $\omega_E = T = \mathrm{conv}\{E,P\}$ shows

$$\|v-v_E\|^2_{L^2(E)} = \frac{|E|}{|\omega_E|} \int_{\omega_E} (v-v_E)^2 \,\mathrm{d}x + \frac{|E|}{n\,|\omega_E|} \int_{\omega_E} (x-P)\cdot\nabla((v-v_E)^2) \,\mathrm{d}x$$

$$\leqslant \frac{|E|}{|\omega_E|}\,\|v - v_E\|^2_{L^2(\omega_E)} + \frac{|E|}{n\,|\omega_E|}\,\operatorname{diam}(\omega_E)\,\big\|\nabla((v-v_E)^2)\big\|_{L^1(\omega_E)}. \quad (3.14)$$

The chain rule and the Poincaré inequality (now with fixed constant $v_E \coloneqq \fint_{\omega_E} v\,\mathrm{d}x$) yields

$$\begin{aligned}\big\|\nabla((v-v_E)^2)\big\|_{L^1(\omega_E)} = \|2(v-v_E)\nabla v\|_{L^1(\omega_E)} &\leqslant 2\,\|v-v_E\|_{L^2(\omega_E)}\,\|\nabla v\|_{L^2(\omega_E)}\\ &\leqslant 2C_{\mathrm{P}}(\omega_E)\,\operatorname{diam}(\omega_E)\,\|\nabla v\|^2_{L^2(\omega_E)}.\end{aligned}$$

Another Poincaré inequality in the first term of (3.14) and the last estimate yield

$$\|v - v_E\|^2_{L^2(E)} \leqslant \frac{\operatorname{diam}(\omega_E)^2}{|\omega_E|}\left(C_{\mathrm{P}}(\omega_E)^2 + 2C_{\mathrm{P}}(\omega_E)/n\right)|E|\,\|\nabla v\|^2_{L^2(\omega_E)}.$$

This concludes the proof for the local estimate on E. A sum over all $E \in \mathcal{E}(\Gamma_N)$ and a Cauchy inequality in $\mathbb{R}^{|\mathcal{E}(\Gamma_N)|}$ lead to the global estimate for $|||\gamma_{\Gamma_N}(g-g^\star)|||_\star$. The proof of the remaining inequality

$$\left\|h_{\mathcal{T}}^{1/2}\lambda_{\min,\mathcal{T}}^{-1/2}(g-g^\star)\right\|_{L^2(E)} \leqslant C_{n-1}\left\|h_{\mathcal{T}}^{1/2}\lambda_{\min,\mathcal{T}}^{-1/2}(g-g_E)\right\|_{L^2(E)}$$

works in the same way as in the first part of the proof with $f^\star$ replaced by $g^\star$ and one dimension less. □

Remark 3.2.3. *(a) Table 3.2 lists some realistic values of the constants of Theorem 3.2.2 in case of triangulations into elements that are shape-similar to the reference element $T_{\mathrm{ref}} \coloneqq \operatorname{conv}\{0, e_1, \ldots, e_n\}$ where e_j is the unit vector in the j-th coordinate direction. Since the constant C_n is quite large, it is advisable to directly compute the norms of $f - f^\star$ and $g - g^\star$.*
(b) Listing 3.2 shows a possible implementation of the evaluation of $\|h_{\mathcal{T}}(f-f^\star)\|_{L^2(\Omega)}$ that employs Listing 3.1 for the dual mesh refinement (Line 5) and the `integrate` *routine of* `AFEM` *to calculate the elementwise constant values of $f^\star$ in Lines 7–10 and the elementwise L^2 norm of $f - f^\star$ in the remaining lines.*

Definition 3.2.4 (Luce-Wohlmuth Equilibration Error Estimator)**.** *The equilibration error estimator after Luce-Wohlmuth employs* (3.6) *for $q = q_{\mathrm{LW}}$ with the properties from Theorem 3.2.2 and reads*

$$\eta^2_{\mathrm{LW}} \coloneqq \sum_{T\in\mathcal{T}}\Bigg(\left\|\mathbb{S}^{-1/2}(q_{\mathrm{LW}} - \sigma_h)\right\|_{L^2(T)} + C_{\mathrm{P}}(T)\left\|\frac{h_T}{\lambda_{\min,T}^{1/2}}(f-f^\star)\right\|_{L^2(T)} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} C_{\mathrm{N}}(E)\left\|\frac{h_E^{1/2}}{\lambda_{\min,T}^{1/2}}(g-g^\star)\right\|_{L^2(E)}\Bigg)^2.$$

The interpolations $f^\star$ and $g^\star$ are defined in (3.9) *and* (3.10) *and the Luce-Wohlmuth equilibrator $q_{\mathrm{LW}} \in \mathrm{RT}_0(\mathcal{T}^\star)$ solves the local problems* (3.11)*.*

Remark 3.2.5 (Comparison with the original Luce-Wohlmuth error estimator)**.** *In comparison to the original error estimator from Luce and Wohlmuth (2004) for $n = 2$ dimensions, the*

```
function val=hotLW(f,c4n,n4e,n4sDb,n4sNb,degree_f)
area4e=computeArea4e(c4n,n4e);
n4s=computeN4s(n4e);
length4s=computeLength4s(c4n,n4s);
s4e=computeS4e(n4e);
[c4n_fine,n4e_fine,n4sDb_fine,n4sNb_fine,parents4e]=refineDual(c4n,n4e,n4sDb,n4sNb);
integrand=@(n4p,pts,pts_ref)(f(pts)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)]);
fstar4e=integrate(c4n,n4e,integrand,degree_f+1);
fstar4e=fstar4e(:,[1 1 2 2 3 3])';
fstar4e=3*fstar4e(:)./area4e(parents4e);
integrand=@(n4p,pts,pts_ref)(f(pts)-fstar4e).^2;
val4e_fine=integrate(c4n_fine,n4e_fine,integrand,degree_f);
val4e=max(length4s(s4e),[],2).^2.*accumarray(parents4e',val4e_fine);
val=sqrt(sum(val4e));
end
```

Listing 3.2: Listing for `hotLW.m`

presented modified design has several advantages which were numerically verified by undisplayed numerical experiments.

(a) For some Dirichlet boundary node $z \in \mathcal{N}(\Gamma_D)$ with $|\partial\omega_z^\star \cap \partial\Gamma_N| = 0$ and adjacent boundary face $E \in \mathcal{E}^\star(\Gamma_D)$, instead of minimising over $Q(\mathcal{T}^\star(z))$, Luce and Wohlmuth fix $q_{\mathrm{LW}} \in Q(\mathcal{T}^\star(z))$ with

$$\int_E q_{\mathrm{LW}} \cdot \nu \,\mathrm{d}s = \frac{1}{2}\left(\int_{\omega_z} f\varphi_z \,\mathrm{d}x - \int_{\partial\omega_z^\star \setminus \partial\Omega} \sigma_h \cdot \nu \,\mathrm{d}s\right) \quad \text{on } E \in \mathcal{E}^\star(\partial\Omega) \cap \mathcal{E}^\star(z).$$

This is suboptimal and the minimisation of the local problem (3.11) *leads to better results, especially on coarse triangulations.*
(b) The upper bound for $|||f + \operatorname{div} q_{\mathrm{LW}}|||_\star$ in the original paper (Luce and Wohlmuth, 2004) involves some constant C_{LW} that results from the approximation and H^1 stability properties of the Scott-Zhang interpolation operator and Poincaré inequalities on patches. It seems unrealistic to assume $C_{LW} \leqslant 1$ and even that choice leads to high contributions on coarse triangulations. The upper bound of $|||f - f^\star|||_\star$ for the modified design is easy to calculate and involves only the known and sharp constants of Theorem 3.2.2.
(c) Another consequence of the present design is the possibility to further modify the fluxes to compute some $q_{\mathrm{LWm}} \in \mathrm{RT}_0(\mathcal{T}^\star)$ with $\operatorname{div} q_{\mathrm{LWm}} + f_{\mathcal{T}^\star} = 0$ where $f_{\mathcal{T}^\star} \in \mathcal{P}_0(\mathcal{T}^\star)$ is the piecewise integral mean of f with respect to the dual mesh $\mathcal{T}^\star$. Since, by the design of $f^\star$,

$$\int_{\partial T} \tau \cdot \nu_T \,\mathrm{d}x = 0 = \int_T f_{\mathcal{T}^\star} - f^\star \,\mathrm{d}x \qquad \text{for all } T \in \mathcal{T},$$

there exists a correction $\tau \in \mathrm{RT}_0(\mathcal{T}^\star)$ with $\tau \cdot \nu_T = 0$ for all $T \in \mathcal{T}$ and $\operatorname{div}\tau = f_{\mathcal{T}^\star} - f^\star$. Then, set $q_{\mathrm{LWm}} = q_{\mathrm{LW}} - \tau$. Figure 3.2 displays the fluxes that are changed by this modification.
A Cauchy inequality in $\mathbb{R}^{|\mathcal{T}^\star(T)|}$ on the subtriangulation $\mathcal{T}^\star(T) \coloneqq \{T^\star \in \mathcal{T}^\star \,|\, T^\star \subset T\}$ leads to

$$\begin{aligned}\int_T (f - f_{\mathcal{T}^\star}) v \,\mathrm{d}x &\leqslant \sum_{T^\star \in \mathcal{T}^\star(T)} C_{\mathrm{P}}(T^\star) \left\| h_{T^\star} \lambda_{\min,T}^{-1/2} (f - f^\star) \right\|_{L^2(T^\star)} \left\| \mathbb{S}^{1/2} \nabla v \right\|_{L^2(T^\star)} \\ &\leqslant C_{\mathrm{P}}(\mathcal{T}^\star(T)) \left\| h_{\mathcal{T}^\star} \lambda_{\min,T}^{-1/2} (f - f^\star) \right\|_{L^2(T)} \left\| \mathbb{S}^{1/2} \nabla v \right\|_{L^2(T)}.\end{aligned}$$

Figure 3.2: Dual refinement $\mathcal{T}^\star(T)$ (left) and red-refinement $\mathrm{red}(T)$ (right) of a triangle $T \in \mathcal{T}$ and modified fluxes in the Luce-Wohlmuth and Braess design for the divergence mean correction after Remark 3.2.5.(c) and Remark 3.2.8.(a), respectively.

The global estimate reads

$$|||f + \operatorname{div} q_{\mathrm{LWm}}|||_\star \leqslant C_{\mathrm{P}}(\mathcal{T}^\star)\,\mathrm{osc}(\lambda_{\min,\mathcal{T}}^{-1/2} f, \mathcal{T}^\star).$$

The oscillations $\mathrm{osc}(f, \mathcal{T}^\star)$ *on* $\mathcal{T}^\star$ *are significantly smaller than* $\mathrm{osc}(f, \mathcal{T})$ *due to the smaller diameter* $h_{\mathcal{T}^\star}$.
In principle, this can be repeated on a further refined mesh, e.g. $\mathrm{red}(\mathcal{T}^\star)$*, and leads to equilibrators whose divergence resolves* f *arbitrarily accurately (up to quadrature errors). On Neumann boundary edges, a similar procedure is possible for the Neumann data* g*. This involves only the adjacent subelements of* $\mathcal{T}^\star$ *for each Neumann boundary edge* $E \in \mathcal{E}(\Gamma_N)$.
(d) The Luce-Wohlmuth equilibration error estimator is equivalent to the residual-error estimator η_{R} *of Subsection 3.4 and this implies efficiency. A proof can be found in Luce and Wohlmuth (2004, Theorem 3.4 on page 1405).*

3.2.3 Design by Braess

The Braess equilibration error estimator is similar to the Luce-Wohlmuth error estimator. The main difference is that the decomposition of the domain employs the partition of unity property of the nodal basis functions and thus leads to overlapping local problems on the node patches of the original triangulation $\mathcal{T}$.

Braess (2007) computes patchwise broken Raviart-Thomas functions

$$q_z := \operatorname*{argmin}_{\tau \in Q(\mathcal{T}(z))} \|\tau\|_{L^2(\omega_z)} \tag{3.15}$$

in the local spaces

$$Q(\mathcal{T}(z)) = \Big\{ \tau \in \varphi_z \sigma_h + \mathrm{RT}_{-1}(\mathcal{T}) \,\Big|\, \operatorname{div}\tau + \fint_T f\varphi_z \,\mathrm{d}x = 0 \text{ on } T \in \mathcal{T}(z),$$
$$\tau\cdot\nu + \fint_E g\varphi_z \,\mathrm{d}x = 0 \text{ on } E \in \mathcal{E}(z) \cap \mathcal{E}(\Gamma_N),\ \tau\cdot\nu = 0 \text{ along } \partial\omega_z \backslash \partial\Omega$$
$$\text{and } \int_E [\tau\cdot\nu_E]_E \,\mathrm{d}s = 0 \text{ on } E \in \mathcal{E}(z)\backslash\mathcal{E}(\Gamma_N) \Big\}.$$

An easy calculation with the partition of unity property shows that the sum

$$q_{\mathrm{B}} \coloneqq \sum_{z\in\mathcal{N}} q_z + \varphi_z\sigma_h = \sigma_h + \sum_{z\in\mathcal{N}} q_z$$

is in $\mathrm{RT}_0(\mathcal{T})$ and satisfies

$$\operatorname{div} q_{\mathrm{B}} + f_{\mathcal{T}} = 0 \text{ in } \Omega \quad \text{and} \quad q\cdot\nu + g_{\mathcal{E}} \text{ along } \Gamma_N.$$

The local terms in (3.6) are bounded by

$$\frac{\int_T (f-f_T)v\,\mathrm{d}x}{\left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}} \leqslant C_{\mathrm{P}}(T)\left\|h_T\lambda_{\min,T}^{-1/2}(f-f_T)\right\|_{L^2(T)} \quad \text{for } T\in\mathcal{T},$$
$$\frac{\int_E (g-g_E)v\,\mathrm{d}x}{\left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}} \leqslant C_{\mathrm{N}}(E)\left\|h_E^{1/2}\lambda_{\min,T}^{-1/2}(g-g_E)\right\|_{L^2(E)} \quad \text{for } E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T).$$

The proof of these estimates is identical to the proof of Theorem 3.2.2. The same holds for the global estimates

$$|||f+\operatorname{div} q_{\mathrm{B}}|||_\star \leqslant C_{\mathrm{P}}(\mathcal{T})\operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2}f,\mathcal{T}),$$
$$|||\gamma_{\Gamma_N}(g-q_{\mathrm{B}}\cdot\nu)|||_\star \leqslant C_{\mathrm{N}}(\mathcal{E}(\Gamma_N))\operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2}g,\mathcal{E}(\Gamma_N)).$$

Lemma 3.2.6. *If $z\in\mathcal{M}$ and $\operatorname{Res}(\varphi_z)=0$, the affine space $Q(\mathcal{T}(z))$ is non-empty. If $z\in\mathcal{N}(\Gamma_D)$, the space $Q(\mathcal{T}(z))$ is non-empty (without further assumptions). For $n=2$ dimensions, the space $Q(\mathcal{T}(z))$ has dimension 1 or 0 depending on the boundary case as given in Table 3.1.*

Proof. The proof is very similar to the one of Lemma 3.2.1 and therefore left to the reader. □

Definition 3.2.7 (Braess Equilibration Error Estimator)**.** *The Braess equilibration error estimator reads*

$$\eta_{\mathrm{B}}^2 \coloneqq \sum_{T\in\mathcal{T}}\Bigg(\left\|\mathbb{S}^{-1/2}(q_{\mathrm{B}}-\sigma_h)\right\|_{L^2(T)} + C_{\mathrm{P}}(T)\left\|h_T\lambda_{\min,T}^{-1/2}(f-f_T)\right\|_{L^2(T)} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} C_{\mathrm{N}}(E)\left\|h_E^{1/2}\lambda_{\min,T}^{-1/2}(g-g_E)\right\|_{L^2(E)}\Bigg)^2.$$

Remark 3.2.8. *(a) Similar to Remark 3.2.5.(c), it is possible to project $q_{\mathrm{B}}\in\mathrm{RT}_0(\mathcal{T})$ onto some $q_{\mathrm{Bm}}\in\mathrm{RT}_0(\operatorname{red}(\mathcal{T}))$ with $\operatorname{div} q_{\mathrm{Bm}} + f_{\operatorname{red}(\mathcal{T})} = 0$ and $q\cdot\nu - g_{\operatorname{red}(\mathcal{E})} = 0$ for the piecewise integral means $f_{\operatorname{red}(\mathcal{T})}\in\mathcal{P}_0(\operatorname{red}(\mathcal{T}))$ of f and $g_{\operatorname{red}(\mathcal{E})}\in\mathcal{P}_0(\operatorname{red}(\mathcal{E}(\Gamma_N)))$ of g with respect to the triangles or Neumann edges of the red-refined triangulation $\operatorname{red}(\mathcal{T})$. The correction modifies only the interior fluxes in $\operatorname{red}(T)$ as displayed in Figure 3.2.*
(b) The Braess equilibration error estimator is equivalent to the residual-based error estimator η_{R} of Subsection 3.4 and this implies efficiency for η_{B}. A proof can be found in Braess (2007, Theorem 9.4 on page 184).

3.2.4 Hyper Circle Identity and MFEM Error Estimator

The hyper circle identity goes back to Prager and Synge (1947) and allows for the comparison of the error of the conforming finite element method with the error of the mixed finite element method. It is a more fundamental approach to equilibration error estimators and implies an efficiency threshold for such estimates based on $\mathrm{RT}_0(\mathcal{T})$ functions.

Theorem 3.2.9 (Hyper Circle Identity)**.** *For any $q \in H(\mathrm{div},\Omega)$, $u, v \in V$ and $\sigma(u) := \mathbb{S}\nabla u \in H(\mathrm{div},\Omega)$ with $\mathrm{div}(q-\sigma) = 0$ and $(q-\sigma)\cdot\nu = 0$, it holds*

$$\left\|\mathbb{S}^{-1/2}(\sigma(v)-q)\right\|^2_{L^2(\Omega)} = |||u-v|||^2 + \left\|\mathbb{S}^{-1/2}(\sigma(u)-q)\right\|^2_{L^2(\Omega)}.$$

Proof. The proof is a simple integration by parts to see that $\int_\Omega \nabla(u-v)(\sigma(u)-q)\,\mathrm{d}x = 0$, see e.g. Braess (2007, Theorem 5.1 on page 148). □

To illustrate an application, set u as the exact solution of the Poisson model problem with exact flux $\sigma := \mathbb{S}\nabla u$ and set $v = u_h$ as the conforming finite element solution with discrete flux $\sigma_h := \mathbb{S}\nabla u_h$. Then, for any $q \in H(\mathrm{div},\Omega)$ with equilibration conditions $\mathrm{div}\, q + f = 0$ in Ω and $q\cdot\nu - g = 0$ along Γ_N,

$$|||\mathrm{Res}|||^2_\star = |||u-u_h|||^2 = \left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|^2_{L^2(\Omega)} - \left\|\mathbb{S}^{-1/2}(\sigma - q)\right\|^2_{L^2(\Omega)}.$$

Nonconstant data and approximated equilibration conditions as for the equilibration error estimators above lead to additional oscillation terms.

Definition 3.2.10 (MFEM Equilibration Error Estimator)**.** *The MFEM equilibrator is the minimiser of*

$$q_{\mathrm{M}} := \operatorname*{argmin}_{\substack{q\in\mathrm{RT}_0(\mathcal{T})\\ \mathrm{div}\, q + f_{\mathcal{T}} = 0 \text{ in } \Omega\\ q\cdot\nu - g_{\mathcal{E}} = 0 \text{ along } \Gamma_N}} \left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|_{L^2(\Omega)}.$$

The MFEM equilibration error estimator reads

$$\eta^2_{\mathrm{M}} := \sum_{T\in\mathcal{T}}\Bigg(\left\|\mathbb{S}^{-1/2}(q_{\mathrm{M}}-\sigma_h)\right\|_{L^2(T)} + C_{\mathrm{P}}(T)\left\|h_T\lambda^{-1/2}_{\min,T}(f-f_T)\right\|_{L^2(T)} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} C_{\mathrm{N}}(E)\left\|h_E^{1/2}\lambda^{-1/2}_{\min,T}(g-g_E)\right\|_{L^2(E)}\Bigg)^2.$$

Remark 3.2.11. *(a) At least for Poisson problems, the MFEM equilibrator equals the solution of the Raviart-Thomas mixed finite element method from Subsection 2.3.3, see Lemma 2.3.4. That is the reason for the nomenclature of this error estimator.*
(b) The error estimator η_{M} is a lower bound for η_{B}, i.e.,

$$|||\mathrm{Res}|||_\star \leqslant \eta_{\mathrm{M}} \leqslant \eta_{\mathrm{B}},$$

because q_{B} satisfies the constraints for the minimisation problem in Definition 3.2.10.

The remaing part of this section derives a threshold that limits the accuracy of equilibration error estimators based on $\mathrm{RT}_0(\mathcal{T})$ equilibrators (such as q_{M} or q_{B} and also q_{LW} if $\mathcal{T}$ is replaced by $\mathcal{T}^\star$). For what follows, we assume constant data f and g (otherwise oscillations appear) for the Poisson model problem.

A compactness argument from Carstensen et al. (2012b) for piecewise constant right-hand sides and $\mathbb{S} = \mathbb{I}$ yields

$$|||u - u_h||| \leqslant C(\mathcal{T}) \left\| \mathbb{S}^{-1/2}(\sigma - q_{\mathrm{M}}) \right\|_{L^2(\Omega)}$$

with some constant $C(\mathcal{T})$, which only depends on the triangulation but neither on the data nor on the exact or discrete solution. Moreover, Theorem 3.2.9 for $q = q_{\mathrm{M}}$ and

$$\kappa \coloneqq \left\| \mathbb{S}^{-1/2}(\sigma - q_{\mathrm{M}}) \right\|_{L^2(\Omega)} / |||u - u_h||| \geqslant 1/C(\mathcal{T})$$

lead to a lower bound for the efficiency index of η_{M} in the sense of

$$\eta_{\mathrm{M}} / |||u - u_h||| \geqslant \sqrt{1 + \kappa^2} \geqslant \sqrt{1 + 1/C(\mathcal{T})}.$$

This marks the hypercircle threshold for the efficiency of η_{M} and other equilibration error estimators (which are upper bounds of η_{M}). Numerical experiments by Carstensen and Merdon (2010, 2013) and in Section 4.3 of this thesis reveal efficiency indices in the range of 1.3 to 1.7 and there is no reason to believe that the threshold is close to one. Carstensen and Merdon (2013) and Section 3.3 below discuss a postprocessing that allows to improve the efficiency of equilibration error estimators below that threshold.

3.2.5 Least-Square FEM and Repin Error Majorants

The theory of error majorants by Repin (2008) is closely related to the least-square finite element method.

Some splits of the integrals in (3.4) lead to

$$\begin{aligned}\int_\Omega (\sigma - g) \cdot \nabla v \,\mathrm{d}x = &\int_\Omega (f - f_{\mathcal{T}}) v \,\mathrm{d}x + \int_\Omega (f_{\mathcal{T}} + \operatorname{div} q) v \,\mathrm{d}x \\ &+ \int_{\Gamma_N} (g - g_{\mathcal{E}}) v \,\mathrm{d}s + \int_{\Gamma_N} (g_{\mathcal{E}} - q \cdot \nu) \,\mathrm{d}s + \int_\Omega (\sigma_h - q) \cdot \nabla v \,\mathrm{d}x.\end{aligned}$$

After Repin (2008), Valdman (2009) or Carstensen and Merdon (2010), this results in the least-square error estimator

$$q_{\mathrm{LS}} \coloneqq \operatorname*{argmin}_{q \in \mathrm{RT}_0(\mathcal{T})} M(q). \tag{3.16}$$

The quantity

$$M(q) \coloneqq \left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|_{L^2(\Omega)} + \frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}}\,\|f_{\mathcal{T}} + \mathrm{div}\, q\|_{L^2(\Omega)} + \frac{C_{\mathrm{T}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)^{1/2}}{\lambda_{\min,\Omega}^{1/2}}\,\|g_{\mathcal{E}} - q\cdot\nu\|_{L^2(\Gamma_N)}$$

is called error majorant by Repin and involves the Friedrichs constant $C_{\mathrm{F}}(\Omega,\Gamma_D)$ from Theorem 2.1.10 and the trace constant $C_{\mathrm{T}}(\Omega,\Gamma_D)$ from Theorem 2.1.11. In practice, the minimum of the sum of norms in η_{LS} can be approximated by a series of least-square problems similar to Valdman (2009). Since $(a+b)^2 = \min_{\lambda>0}\left((1+1/\lambda)a^2 + (1+\lambda)b^2\right)$ with minimiser $\lambda = b/a$, the majorant equals

$$M(q)^2 = \min_{\lambda>0}\widehat{M}(\lambda,q) \qquad \text{with}$$

$$\widehat{M}(\lambda,q) \coloneqq (1+\lambda)\left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|^2_{L^2(\Omega)} + (1+\frac{1}{\lambda})\left(\frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}}\,\|f_{\mathcal{T}} + \mathrm{div}\, q\|_{L^2(\Omega)} + \frac{C_{\mathrm{T}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)^{1/2}}{\lambda_{\min,\Omega}^{1/2}}\,\|g_{\mathcal{E}} - q\cdot\nu\|_{L^2(\Gamma_N)}\right)^2.$$

So one may fix $\lambda = 1$ and minimise $\widehat{M}(\lambda,q)$ over $q \in \mathrm{RT}_0(\mathcal{T})$ and then update λ with

$$\lambda = \frac{\frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}}\,\|f_{\mathcal{T}} + \mathrm{div}\, q\|_{L^2(\Omega)} + \frac{C_{\mathrm{T}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)^{1/2}}{\lambda_{\min,\Omega}^{1/2}}\,\|g_{\mathcal{E}} - q\cdot\nu\|_{L^2(\Gamma_N)}}{\left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|_{L^2(\Omega)}}.$$

This concept naturally extends to another minimisation of the sum of the two norms

$$\frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}}\,\|f_{\mathcal{T}} + \mathrm{div}\, q\|_{L^2(\Omega)} + \frac{C_{\mathrm{T}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)^{1/2}}{\lambda_{\min,\Omega}^{1/2}}\,\|g_{\mathcal{E}} - q\cdot\nu\|_{L^2(\Gamma_N)}.$$

Alternatively, as done in the implementations for this thesis, we can restrict the minimisation to $q \in \mathrm{RT}_0(\mathcal{T})$ with $q\cdot\nu - g_{\mathcal{E}} = 0$ along Γ_N such that the latter term vanishes. This may be suboptimal, but circumvents the problem of the knowledge of $C_{\mathrm{T}}(\Omega,\Gamma_D)$. Algorithm 3.1 applies three iterations of the strategy described above and so is near the optimal solution. Since λ may converge to 0, the linear system can get singular or ill-conditioned. In this case, the algorithm stops and continues with the last iterate.

The combination of q_{LS} with (3.6) leads to the Definition 3.2.12 below with the additional term

$$\sup_{v\in V}\frac{\int_\Omega (f_{\mathcal{T}} + \mathrm{div}\, q_{\mathrm{LS}})v\,\mathrm{d}x}{|||v|||} \leqslant \frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}}\,\|f_{\mathcal{T}} + \mathrm{div}\, q_{\mathrm{LS}}\|_{L^2(\Omega)}$$

which allows no local contribution to the sum over $T \in \mathcal{T}$ but vanishes if $\mathrm{div}\, q_{\mathrm{LS}} + f_{\mathcal{T}} = 0$.

Set $\lambda \coloneqq 1$;
for $\ell \coloneqq 1, 2, 3$ **do**
  Compute $q_{\mathrm{LS}} \coloneqq \operatorname{argmin}\left\{\widehat{M}(\lambda, q) \,\middle|\, q \in \mathrm{RT}_0(\mathcal{T}),\ q \cdot \nu - g_{\mathcal{E}} \text{ along } \Gamma_N\right\}$;
  Update $\lambda \coloneqq \left(\frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}} \|f_{\mathcal{T}} + \operatorname{div} q\|_{L^2(\Omega)}\right) / \left\|\mathbb{S}^{-1/2}(\sigma_h - q)\right\|_{L^2(\Omega)}$;
  if *linear system is nearly singular* **then**
    break;
Output: q_{LS}

Algorithm 3.1: Algorithm for approximation of the least-square error estimator.

Definition 3.2.12 (Least-Square Equilibration Error Estimator)**.** *The least-square equilibration error estimator for q_{LS} from Algorithm 3.1 reads*

$$\eta_{\mathrm{LS}} \coloneqq \frac{C_{\mathrm{F}}(\Omega,\Gamma_D)\,\mathrm{diam}(\Omega)}{\lambda_{\min,\Omega}^{1/2}} \|f_{\mathcal{T}} + \operatorname{div} q_{\mathrm{LS}}\|_{L^2(\Omega)} + \left(\sum_{T\in\mathcal{T}}\left(\left\|\mathbb{S}^{-1/2}(q_{\mathrm{LS}} - \sigma_h)\right\|_{L^2(T)}\right.\right.$$
$$\left.\left. + C_{\mathrm{P}}(T)\left\|h_T \lambda_{\min,T}^{-1/2}(f - f_T)\right\|_{L^2(T)} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} C_{\mathrm{N}}(E)\left\|h_E^{1/2}\lambda_{\min,T}^{-1/2}(g - g_E)\right\|_{L^2(E)}\right)^2\right)^{1/2}.$$

Remark 3.2.13 (Comparison with η_{M})**.** *Since q_{M} is an admissible quantity in the minimisation of $M(q)$ in* (3.16)*, it holds $\eta_{\mathrm{LS}} \leqslant \eta_{\mathrm{M}}$. However, supercloseness results in Brandts et al. (2006) suggest also*

$$\eta_{\mathrm{M}} \leqslant \eta_{\mathrm{LS}} + \textit{terms of higher order}.$$

So asymptotically both error estimators coincide, but η_{LS} might lead to some improvement on coarse triangulations. All experiments for Poisson problems in Section 4.3 verify that the approximation of η_{LS} by Algorithm 3.1 shows the asymptotical coincidence, but on coarse grids the computed values of η_{LS} often fail to improve η_{M} (even if the number of iterations in Algorithm 3.1 is increased). Moreover, for guaranteed error control the constants $C_{\mathrm{F}}(\Omega,\Gamma_D)$ and $C_{\mathrm{T}}(\Omega,\Gamma_D)$ (or upper bounds of them) have to be incorporated. Altogether, it appears reasonable to prefer η_{M} over η_{LS}.

3.3 Effective Postprocessing for Equilibration Error Estimators

This section introduces the novel postprocessing for equilibration a posteriori error estimators published by Carstensen and Merdon (2013). It is applicable to all equilibration designs, e.g. the ones that are discussed above and summarised in Table 3.3.

3.3.1 Motivation and Asymptotic Exactness

Consider some residual (3.1) based on the data $f \in L^2(\Omega)$, $g \in L^2(\Gamma_N)$, $\sigma_h \in L^2(\Omega;\mathbb{R}^n)$, and some (equilibrated) quantity $q \in H(\operatorname{div},\Omega)$ (e.g. the ones from Table 3.3). Then, for

η	Equilibration Conditions	Details
η_{LW}	$\operatorname{div} q_{\mathrm{LW}} + f^\star = 0,\ q_{\mathrm{LW}} \cdot \nu - g^\star = 0$	Subsection 3.2.2
η_{LWm}	$\operatorname{div} q_{\mathrm{LWm}} + f_{\mathcal{T}^\star} = 0,\ q_{\mathrm{LWm}} \cdot \nu - g_{\mathcal{E}(\Gamma_N)^\star} = 0$	Remark 3.2.5.(c)
η_{B}	$\operatorname{div} q_{\mathrm{B}} + f_{\mathcal{T}} = 0,\ q_{\mathrm{B}} \cdot \nu - g_{\mathcal{E}(\Gamma_N)} = 0$	Subsection 3.2.3
η_{Bm}	$\operatorname{div} q_{\mathrm{Bm}} + f_{\mathrm{red}^j(\mathcal{T})} = 0,\ q_{\mathrm{Bm}} \cdot \nu - g_{\mathrm{red}^j(\mathcal{E}(\Gamma_N))} = 0$	Remark 3.2.8.(a)
η_{M}	$\operatorname{div} q_{\mathrm{M}} + f_{\mathcal{T}} = 0,\ q_{\mathrm{M}} \cdot \nu - g_{\mathcal{E}(\Gamma_N)} = 0$	Subsection 3.2.4
η_{LS}	none	Subsection 3.2.5

Table 3.3: Equilibration a posteriori error estimators suitable for postprocessing.

all $\varphi \in V$ and $v \in H^s(\Omega)$ with $\operatorname{Curl} v \cdot \nu = 0$ along Γ_N, an integration by parts leads to

$$\operatorname{Res}(\varphi) = \int_\Omega (f + \operatorname{div} q)\varphi \,\mathrm{d}x + \int_{\Gamma_N} (g - q \cdot \nu)\varphi \,\mathrm{d}s + \int_\Omega (q - \sigma_h - \operatorname{Curl} v) \cdot \nabla\varphi \,\mathrm{d}x.$$

The Helmholtz decomposition (Theorem 2.1.12) of $q - \sigma_h$ yields

$$q - \sigma_h = \mathbb{S}\nabla a + \operatorname{Curl} b \tag{3.17}$$

with a unique $a \in H^1_D(\Omega)$ and remainder $b \in H^s(\Omega)$ with $\operatorname{Curl} b \cdot \nu = 0$ along Γ_N. The optimal postprocessing of

$$\begin{aligned}\eta &\coloneqq |||f + \operatorname{div} q|||_\star + |||\gamma_{\Gamma_N}(g - q \cdot \nu)|||_\star + \left\|\mathbb{S}^{-1/2}(q - \sigma_h)\right\|_{L^2(\Omega)} \\ &= |||f + \operatorname{div} q|||_\star + |||\gamma_{\Gamma_N}(g - q \cdot \nu)|||_\star + \left(|||a|||^2 + \left\|\mathbb{S}^{-1/2}\operatorname{Curl} b\right\|^2_{L^2(\Omega)}\right)^{1/2}\end{aligned}$$

with $v = b$ results in

$$\mu \coloneqq |||f + \operatorname{div} q|||_\star + |||\gamma_{\Gamma_N}(g - q \cdot \nu)|||_\star + |||a||| \leqslant \eta.$$

With the assumption that

$$\delta \coloneqq |||f + \operatorname{div} q|||_\star + |||\gamma_{\Gamma_N}(g - q \cdot \nu)|||_\star \quad \text{is of higher order,} \tag{3.18}$$

the subsequent Theorem 3.3.1 implies asymptotic exactness in the sense that

$$\mu/(1 + 2\delta) \leqslant |||\operatorname{Res}|||_\star \leqslant \mu.$$

Theorem 3.3.1. *With the notation from* (3.17) *and* (3.18)*, it holds*

$$\begin{aligned}&|||\operatorname{Res}|||_\star \leqslant \eta \leqslant \left((|||\operatorname{Res}|||_\star + \delta)^2 + \left\|\mathbb{S}^{-1/2}\operatorname{Curl} b\right\|^2_{L^2(\Omega)}\right)^{1/2} + \delta, \\ &|||\operatorname{Res}|||_\star \leqslant \mu \leqslant |||\operatorname{Res}|||_\star + 2\delta.\end{aligned}$$

For $|||a||| > 0$ *and* $\kappa := \left\|\mathbb{S}^{-1/2}\operatorname{Curl} b\right\|_{L^2(\Omega)} / |||a|||$, *it holds*

$$0 \leqslant \eta - \mu = |||a||| \left(\sqrt{1+\kappa^2} - 1\right) \leqslant (\delta + |||\mathrm{Res}|||_\star) \left(\sqrt{1+\kappa^2} - 1\right).$$

Proof. The Helmholtz decomposition (3.17) shows, for all $\varphi \in H^1_D(\Omega)$,

$$\begin{aligned}\mathrm{Res}(\varphi) &= \int_\Omega (f + \operatorname{div} q)\varphi \,\mathrm{d}x + \int_{\Gamma_N} (g - \sigma_h \cdot \nu)\varphi \,\mathrm{d}x + \int_\Omega (q - \sigma_h - \operatorname{Curl} b) \cdot \nabla\varphi \,\mathrm{d}x \\ &\leqslant (\delta + |||a|||)\, |||\varphi|||\,.\end{aligned}$$

Hence,

$$|||\mathrm{Res}|||_\star \leqslant \delta + |||a||| = \mu \leqslant \eta = \delta + \sqrt{|||a|||^2 + \left\|\mathbb{S}^{-1/2}\operatorname{Curl} b\right\|^2_{L^2(\Omega)}}.$$

The improvement factor of the second term is

$$|||a||| \,/ \sqrt{|||a|||^2 + \left\|\mathbb{S}^{-1/2}\operatorname{Curl} b\right\|^2_{L^2(\Omega)}} = 1/\sqrt{1+\kappa^2}.$$

Moreover,

$$\begin{aligned}|||a|||^2 &= \int_\Omega \mathbb{S}\nabla a \cdot \nabla a \,\mathrm{d}x = \int_\Omega (q - \sigma_h) \cdot \nabla a \,\mathrm{d}x \\ &= \mathrm{Res}(a) - \int_\Omega (f + \operatorname{div} q) a \,\mathrm{d}x - \int_{\Gamma_N} (g - \sigma_h \cdot \nu) a \,\mathrm{d}x \\ &\leqslant (|||\mathrm{Res}|||_\star + \delta)\, |||a|||\,.\end{aligned}$$

This concludes the proof. □

3.3.2 Algorithmic Realisation

The algorithmic realisation of the postprocessing employs continuous and piecewise affine functions $v \in \mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega)$ on some (possibly refined) triangulation $\widehat{\mathcal{T}}$ of Ω, e.g. $\widehat{\mathcal{T}} \in \{\mathcal{T}, \mathcal{T}^\star, \mathrm{red}(\mathcal{T}), \mathrm{red}(\mathcal{T}^\star), \mathrm{red}^2(\mathcal{T}), \ldots\}$. The optimal postprocessing $v \in \mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega)$ within this discrete space reads

$$v := \operatorname*{argmin}_{\substack{\gamma \in \mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega) \\ \operatorname{Curl}\gamma \cdot \nu = 0 \text{ along } \Gamma_N}} \left\|\mathbb{S}^{-1/2}(\sigma_h - q - \operatorname{Curl}\gamma)\right\|_{L^2(\Omega)}. \tag{3.19}$$

For an equilibrated quantity $q \in \mathrm{RT}_0(\widehat{\mathcal{T}})$ on some triangulation $\widehat{\mathcal{T}}$, the minimisation of the right-hand side amongst $\gamma \in \mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega)$ results in some linear system of equations `Ax = b` for the coefficient vector `x` with respect to some basis $\{\varphi_{z_1}, \ldots, \varphi_{z_N}\}$ of

$\mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega)$. The entries of the stiffness matrix A and the right-hand side vector b read

$$A_{jk} \coloneqq \int_\Omega \mathbb{S}^{-1} \operatorname{Curl} \varphi_{z_j} \cdot \operatorname{Curl} \varphi_{z_k} dx \quad \text{and} \quad b_j \coloneqq \int_\Omega \mathbb{S}^{-1}(\sigma_h - q) \cdot \operatorname{Curl} \varphi_{z_j} \, \mathrm{d}x.$$

The MATLAB routine `pcg` with (diagonal) Jacobi preconditioner $D = \operatorname{diag}(A_{11}, \ldots, A_{NN})$ and initial value $x_0 = 0$ solves this system iteratively in k iterations with the first iterate

$$x_1 \coloneqq \frac{b^T D^{-1/2} b}{b^T D^{-1/2} A D^{-1/2} b} D^{-1/2} b.$$

η	Original Equilibrator q	Reference for q	Mesh $\widehat{\mathcal{T}}$
$\eta_{\mathrm{LW(k)}}$	q_{LW}	Definition 3.2.4	$\mathcal{T}^\star$
$\eta_{\mathrm{LWm(k)}}$	q_{LWm}	Remark 3.2.5.(c)	$\mathcal{T}^\star$
$\eta_{\mathrm{Br(k)}}$	q_{B}	Definition 3.2.7	$\operatorname{red}(\mathcal{T})$
$\eta_{\mathrm{Bmr(k)}}$	q_{Bm}	Remark 3.2.8.(a)	$\operatorname{red}(\mathcal{T})$
$\eta_{\mathrm{Brr(k)}}$	q_{B}	Definition 3.2.7	$\operatorname{red}^2(\mathcal{T})$
$\eta_{\mathrm{Bmrr(k)}}$	q_{Bm}	Remark 3.2.8.(a)	$\operatorname{red}^2(\mathcal{T})$

Table 3.4: List of postprocessed equilibration a posteriori error estimators, the original equilibrator they are based on and the mesh $\widehat{\mathcal{T}}$ that is employed for the postprocessing with k pcg iterations.

Since $\operatorname{Curl}(\mathcal{P}_1(\mathcal{T}) \cap C(\Omega)) \subset \mathrm{RT}_0(\mathcal{T})$ and $\left\|\mathbb{S}^{-1/2}(\sigma_h - q_{\mathrm{M}})\right\|_{L^2(\Omega)}$ is already the best-approximation in $\mathrm{RT}_0(\mathcal{T})$, there is no improvement by the postprocessing with $\widehat{\mathcal{T}} = \mathcal{T}$ in case of the MFEM error estimator η_{M}, unless one refines the mesh or increases the polynomial degree. But the postprocessing with $\widehat{\mathcal{T}} = \mathcal{T}$ may reduce the gap between η_{M} and the Braess equilibration error estimator η_{B}. We suggest to use $\widehat{\mathcal{T}} = \mathcal{T}^\star$ for the Luce-Wohlmuth error estimator and a red-refinement $\widehat{\mathcal{T}} = \operatorname{red}(\mathcal{T})$ for all other error estimators, see also Table 3.4. The postprocessed quantity $q_{\mathrm{xyz}} - \operatorname{Curl} v$ replaces the original equilibrator q_{xyz} in the corressponding Definition of η_{xyz} to define the new error estimators from Table 3.4.

The Luce-Wohlmuth error estimator with postprocessing $\widehat{\mathcal{T}} = \mathcal{T}^\star$ and k iterations is labelled as $\eta_{\mathrm{LW(k)}}$. The Braess error estimator with postprocessing on $\widehat{\mathcal{T}} = \mathcal{T}$ and k iterations is labelled as $\eta_{\mathrm{B(k)}}$. For $\widehat{\mathcal{T}} = \operatorname{red}(\mathcal{T})$ and k iterations, we add a subscript "r" in the label, e.g. $\eta_{\mathrm{Br(k)}}$ in case of the Braess error estimator. For every additional red-refinement, another "r" is added. For instance, $\eta_{\mathrm{Brr(3)}}$ implies $\widehat{\mathcal{T}} = \operatorname{red}^2(\mathcal{T})$ and $k = 3$ cg iterations. Of course, the postprocessing can also be combined with the mean-corrected versions η_{LWm} and η_{Bm} suggested in Remark 3.2.5.(c) and Remark 3.2.8.(a). In case of two red-refinements $\widehat{\mathcal{T}} = \operatorname{red}^2(\mathcal{T})$, the mean correction is executed twice.

Remark 3.3.2. *(a) Further red-refinements $\widehat{\mathcal{T}} = \operatorname{red}^j(\mathcal{T})$ for $j \geqslant 2$ lead to even more accurate guaranteed upper bounds and satisfy a reduction property derived in Carstensen and Merdon (2013) for model scenarios. For instance, for the Braess equilibration error estimator, there exist constants $0 < \varrho < 1$ and $0 < \Lambda < \infty$, which depend on the interior angles of $\mathcal{T}$ and neither on*

the mesh sizes nor on the number of elements, such that

$$\eta^2_{\mathrm{Br}(\infty)} - |||e|||^2 \leqslant \varrho \left(\eta^2_{\mathrm{B}} - |||e|||^2 \right) + \Lambda \operatorname{Osc}^2(f, \mathcal{T})$$

with the edge-patch-related big oscillations

$$\operatorname{Osc}^2(f, \mathcal{T}) \coloneqq \sum_{E \in \mathcal{E}(\Omega)} \operatorname{diam}(\omega_E)^2 \, \|f - f_{\omega_E}\|^2_{L^2(\omega_E)} + \sum_{E \in \mathcal{E}(\partial\Omega)} |E|^2 \, \|f\|^2_{L^2(\omega_E)} \,.$$

This reduction property can be iterated for $\eta_{\mathrm{Brr}(\infty)}$ and the postprocessings based on more red-refinements. The numerical examples by Carstensen and Merdon (2013) suggest that ϱ is about 0.3 (for $k = \infty$) and about 0.5 for the cheaper version with $k = 1$ iterations.
(b) In the 3D case, the minimisation problem (3.19) *involves the* $\operatorname{Curl} \coloneqq \nabla \times \psi$ *of functions ψ in $H^1(\Omega; \mathbb{R}^3)$. This causes modifications in the realisation of the postprocessing, either by the choice of a proper basis of $P^1(\Omega; \mathbb{R}^3) \cap C(\Omega)$ or by $H(\operatorname{curl}, \Omega)$-conforming finite elements.*

3.3.3 Implementation Issues

In the 2D implementation for this thesis, Raviart-Thomas elements are represented by their three normal fluxes on every triangle which are the coefficients for the three basis functions ϑ from Theorem 2.2.7. An array, denoted `quh` in the code, contains these fluxes in the following convention. The entry `quh(j,k)` contains the normal flux of the k-th edge (with respect to the numbering induced by `s4e`, see Subsubsection 2.3.5.2 for details) on the j-th triangle of $\mathcal{T}$.

Listing 3.3 shows the `MATLAB` code for the function `RT2RedRT` that projects the degrees of freedom of some Raviart-Thomas element in $\mathrm{RT}_0(\mathcal{T})$ to the degrees of freedom with respect to $\mathrm{RT}_0(\operatorname{red}(\mathcal{T}))$. Lines 12–17 evaluate the basis functions on the new edge midpoints in the red-refined triangulation $\operatorname{red}(\mathcal{T})$ by `getRTBasis` from Listing 2.4 and Lines 18–20 compute their normal fluxes.

The remaining lines compute the mean correction of the divergence as described in Remark 3.2.8.(a). The mean correction of the normal fluxes along the Neumann boundary was also implemented but is not displayed here.

3.4 Explicit Residual-Based Error Estimator

This section concerns the standard explicit residual-based error estimator. The reliability proof usually employs stability and approximation properties of Clément interpolation operators, see e.g. (Carstensen, 1999; Funken, 2002). In case of discontinuous coefficients $\mathrm{S} \in \mathcal{P}_0(\mathcal{T}; \mathbb{R}^{n \times n})$ the reliability constants involve eigenvalues like $\lambda_{\max,\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$ from (3.2). Under certain assumptions it is possible to derive constants that do not depend on the ratio between the smallest and largest eigenvalue, cf. Bernardi and Verfürth (2000) for details.

Definition 3.4.1 (Standard Residual Error Estimator)**.** *The standard residual-based error*

```
function [c4n_fine,n4e_fine,n4sDb_fine,n4sNb_fine,parents4e,quh_fine]=...
    RT2RedRT(c4n,n4e,n4sDb,n4sNb,quh,f,degree_f)
[c4n_fine,n4e_fine,n4sDb_fine,n4sNb_fine,parents4e,~,~,~,childnr]=...
    refineUniformRed(c4n,n4e,n4sDb,n4sNb);
quh_fine=zeros(size(n4e_fine,1),3);
n4s_fine=computeN4s(n4e_fine);
s4e_fine=computeS4e(n4e_fine);
mid4s_fine=computeMid4s(c4n_fine,n4s_fine);
length4s=computeLength4s(c4n_fine,n4s_fine);
normal4e_fine=computeNormal4e(c4n_fine,n4e_fine);
area4e=computeArea4e(c4n,n4e);
val_on_edge1=getRTBasis(mid4s_fine(s4e_fine(:,1),:),c4n,n4e(parents4e,:),area4e(parents4e));
val_on_edge2=getRTBasis(mid4s_fine(s4e_fine(:,2),:),c4n,n4e(parents4e,:),area4e(parents4e));
val_on_edge3=getRTBasis(mid4s_fine(s4e_fine(:,3),:),c4n,n4e(parents4e,:),area4e(parents4e));
val_on_edge1=matMul(val_on_edge1,quh(parents4e,:,:));
val_on_edge2=matMul(val_on_edge2,quh(parents4e,:,:));
val_on_edge3=matMul(val_on_edge3,quh(parents4e,:,:));
quh_fine(:,1)=sum(val_on_edge1.*permute(normal4e_fine(1,:,:),[3 2 1]),2).*length4s(s4e_fine(:,1));
quh_fine(:,2)=sum(val_on_edge2.*permute(normal4e_fine(2,:,:),[3 2 1]),2).*length4s(s4e_fine(:,2));
quh_fine(:,3)=sum(val_on_edge3.*permute(normal4e_fine(3,:,:),[3 2 1]),2).*length4s(s4e_fine(:,3));
if nargin>6
    mean4e_coarse=sum(quh,2);
    if nargin(f_fine)==3
        integrand=@(n4p,pts,pts_ref)(f_fine(n4p,pts,pts_ref));
    else
        integrand=@(n4p,pts,pts_ref)(f_fine(pts));
    end
    mean4e_fine=integrate(c4n_fine,n4e_fine,integrand,degree_f);
    childs1=find(childnr == 1);
    corr1=mean4e_coarse(parents4e(childs1))/4-mean4e_fine(childs1);
    childs2=find(childnr == 2);
    corr2=mean4e_coarse(parents4e(childs2))/4-mean4e_fine(childs2);
    childs3=find(childnr == 3);
    corr3=mean4e_coarse(parents4e(childs3))/4-mean4e_fine(childs3);
    quh_fine(childs1,2)=quh_fine(childs1,2)-corr1;
    quh_fine(childs2,3)=quh_fine(childs2,3)-corr2;
    quh_fine(childs3,1)=quh_fine(childs3,1)-corr3;
    childs4=find(childnr==4);
    quh_fine(childs4,1)=quh_fine(childs4,1)+corr3;
    quh_fine(childs4,2)=quh_fine(childs4,2)+corr1;
    quh_fine(childs4,3)=quh_fine(childs4,3)+corr2;
    % Neumann flux mesan correction not displayed
end
```

Listing 3.3: Listing for `RT2RedRT.m`

estimator reads

$$\eta_{\mathrm{R}} \coloneqq \left(\sum_{T\in\mathcal{T}} \frac{h_T^2}{\lambda_{\max,T}} \|f\|_{L^2(\Omega)}^2 \right)^{1/2} + \left(\sum_{E\in\mathcal{E}} \frac{h_E}{\lambda_{\max,\omega_E}} \|[\sigma_h \cdot \nu_E]_E\|_{L^2(E)}^2 \right)^{1/2}$$

with the side-based diffusion coefficients $\lambda_{\max,\omega_E} \coloneqq \max_{T\in\mathcal{T}(E)} \lambda_{\max,T}$ *and the jumps of Definition 3.1.1.*

3.4.1 Novel Reliability Proof with Explicit Constants

The recent ansatz from Carstensen and Merdon (2013+) for the proof of reliability of the residual error estimator η_{R} in the sense of

$$|||\mathrm{Res}|||_\star \leqslant C_{\mathrm{rel}}\eta_{\mathrm{R}} + \mathrm{hot}$$

employs the equilibrated fluxes $q_{\mathrm{LW}} \in \mathrm{RT}_0(\mathcal{T}^\star)$ on the dual mesh $\mathcal{T}^\star$ of Subsection 3.2.2. The outcome is a novel explicit residual-based error estimator

$$|||\mathrm{div}(q_{\mathrm{LW}} - \sigma_h)|||_\star \leqslant \eta_{\mathrm{R}}^\star \coloneqq \left(\sum_{z\in\mathcal{N}} \big(c_1(z)\eta(z) + c_2(z)\eta(\mathcal{E}(z))\big)^2 \right)^{1/2} \approx \eta_{\mathrm{R}}$$

with constants $c_1(z)$ and $c_2(z)$ in front of

$$\eta(z) \coloneqq \mathrm{diam}(\omega_z^\star)\lambda_{\min,\omega_z^\star}^{-1/2} \left\| \varphi_z^{1/2}(f - f_{\omega_z}) \right\|_{L^2(\omega_z)} \quad \text{and}$$

$$\eta(\mathcal{E}(z)) \coloneqq \lambda_{\min,\omega_z^\star}^{-1/2} \left(\sum_{E\in\mathcal{E}(z)} |E|^{1/(n-1)} \left\| \varphi_z^{1/2}[\sigma_h \cdot \nu_E]_E \right\|_{L^2(E)}^2 \right)^{1/2} \quad \text{for each } z \in \mathcal{N}.$$

In the following analysis the Poincaré-Friedrichs constants play a dominant role. They read

$$C_{\mathrm{PF}}(\omega_z^\star) \coloneqq \begin{cases} \sup\left\{ \|f\|_{L^2(\omega_z^\star)} \,:\, f \in V \;\&\; \|\nabla f\|_{L^2(\omega_z^\star)} = 1 \right\} & \text{for } z \in \mathcal{N}(\Gamma_D), \\ \sup\left\{ \left\| f - \fint_{\omega_z^\star} f \,\mathrm{d}x \right\|_{L^2(\omega_z^\star)} \,:\, f \in V \;\&\; \|\nabla f\|_{L^2(\omega_z^\star)} = 1 \right\} & \text{for } z \in \mathcal{M}. \end{cases}$$

Notice that these constants include the dependency on the diameter or width of $\omega_z^\star$ in contrast to the constants from Theorems 2.1.8 or 2.1.10. Definition 3.1.1 of the jump $[\sigma_h \cdot \nu_E]_E$ of σ_h over some side $E \in \mathcal{E}$ is slightly extended for Neumann sides of the dual triangulation, i.e.,

$$[\sigma_h \cdot \nu_E]_E \coloneqq \sigma_h \cdot \nu - g^\star \quad \text{for } E \in \mathcal{E}^\star(\Gamma_N).$$

Theorem 3.4.2. *Any* $\sigma_h^\star \in \mathrm{RT}_0(\mathcal{T}^\star)$ *with* $(\sigma_h^\star - \sigma_h) \cdot \nu = 0$ *along* $\partial\omega_z^\star \backslash \partial\Omega$, $\mathrm{div}\,\sigma_h^\star + f^\star = 0$ *and* $\sigma_h^\star \cdot \nu - g = 0$ *along* Γ_N *(e.g.* $\sigma_h^\star = q_{\mathrm{LW}}$ *from Subsection 3.2.2) satisfies*

$$|||\mathrm{div}(\sigma_h^\star - \sigma_h)|||_\star^2 \leqslant \sum_{z\in\mathcal{N}} \big(c_1(z)\eta(z) + c_2(z)\eta(\mathcal{E}(z))\big)^2 .$$

The constants are bounded by

$$c_1(z) \leqslant C_{\mathrm{PF}}(\omega_z^\star)/\operatorname{diam}(\omega_z^\star) \lesssim 1,$$

$$c_2(z) \leqslant \frac{\sqrt{2n}}{(n!)^{n/(2(n-1))}} \max_{F\in\mathcal{F}(z)} |F|^{\frac{1}{2}-\frac{1}{2(n-1)}} \left(C_{\mathrm{PF}}(\omega_z^\star)^2 \, |\omega_F^\star|^{-1} + \frac{\|\bullet - \operatorname{mid}(\mathcal{T})\|^2_{L^2(\omega_F^\star)}}{n^2 \, |\omega_F^\star|^2} \right)^{1/2} \lesssim 1$$

with $\mathcal{F}(z) := \{F \in \mathcal{E}^\star(z) \,\big|\, F \subseteq \bigcup(\mathcal{E}(z)\backslash\mathcal{E}(\Gamma_D))\}$.

Proof. Given any $v \in V$, set $v_{\omega_z^\star} = \int_{\omega_z^\star} v \,\mathrm{d}x / |\omega_z^\star|$ in case of a free node $z \in \mathcal{M}$ and set $v_{\omega_z^\star} := 0$ in case of a Dirichlet boundary node $z \in \mathcal{N}(\Gamma_D)$. Since $(\sigma_h^\star - \sigma_h)\cdot\nu = 0$ along $\partial\omega_z^\star\backslash\partial\Omega$, an integration by parts shows

$$\begin{aligned}\int_{\omega_z^\star} (\sigma_h^\star - \sigma_h)\cdot\nabla v \,\mathrm{d}x &= \int_{\omega_z^\star} (\sigma_h^\star - \sigma_h)\cdot\nabla(v - v_{\omega_z^\star}) \,\mathrm{d}x \\ &= -\int_{\omega_z^\star} (v - v_{\omega_z^\star}) \operatorname{div}\sigma_h^\star \,\mathrm{d}x - \sum_{F\in\mathcal{E}^\star(z)} \int_F (v - v_{\omega_z^\star})[\sigma_h\cdot\nu_F]_F \,\mathrm{d}s \\ &\quad + \int_{\partial\omega_z^\star\cap\partial\Omega} (v - v_{\omega_z^\star})(\sigma_h^\star - \sigma_h)\cdot\nu \,\mathrm{d}s =: \mathrm{I} + \mathrm{II} + \mathrm{III}.\end{aligned}$$

Since $v = v_{\omega_z^\star} = 0$ along $\partial\omega_z^\star \cap \partial\Omega$, the third expression $\mathrm{III} = 0$ vanishes. Since $\int_{\omega_z^\star}(v - v_{\omega_z^\star}) \,\mathrm{d}x = 0$ in case of $z \in \mathcal{N}(\Omega)$, one can add f_{ω_z} in the first expression I. Then, the Poincaré or Friedrichs inequality yields

$$\mathrm{I} := -\int_{\omega_z^\star} (v - v_{\omega_z^\star})(\operatorname{div}\sigma_h^\star + f_{\omega_z}) \,\mathrm{d}x \leqslant C_{\mathrm{PF}}(\omega_z^\star) \, \|\operatorname{div}\sigma_h^\star + f_{\omega_z}\|_{L^2(\omega_z^\star)} \, \|\nabla v\|_{L^2(\omega_z^\star)}.$$

For each (of the $n!$ subelements) $T^\star \in \mathcal{T}^\star(z)$ with $\mathcal{N}(T)\cap\mathcal{N}^\star(T^\star) = \{z\}$, $\operatorname{div}\sigma_h^\star + f^\star = 0$ leads to

$$\begin{aligned}\|\operatorname{div}\sigma_h^\star + f_{\omega_z}\|^2_{L^2(T^\star)} = \|f^\star - f_{\omega_z}\|^2_{L^2(T^\star)} &= (n+1)^2 \, |T^\star| / |T|^2 \left| \int_T (f - f_{\omega_z})\varphi_z \,\mathrm{d}x \right|^2 \\ &\leqslant \frac{(n+1)^2}{(n+1)!} \, |T|^{-1} \left\| \varphi_z^{1/2} \right\|^2_{L^2(T)} \left\| \varphi_z^{1/2}(f - f_{\omega_z}) \right\|^2_{L^2(T)}.\end{aligned}$$

Since $\left\| \varphi_z^{1/2} \right\|^2_{L^2(T)} = |T|/(n+1)$ (see Lemma 2.2.18), this proves

$$\|f^\star - f_{\omega_z}\|^2_{L^2(T^\star)} \leqslant \frac{1}{n!} \left\| \varphi_z^{1/2}(f - f_{\omega_z}) \right\|^2_{L^2(T)}. \tag{3.20}$$

Since each $T \in \mathcal{T}(z)$ contains $n!$ subelements $T^\star \in \mathcal{T}^\star(z)$,

$$\|f^\star - f_{\omega_z}\|^2_{L^2(\omega_z^\star)} \leqslant \sum_{T\in\mathcal{T}(z)} \left\| \varphi_z^{1/2}(f - f_{\omega_z}) \right\|^2_{L^2(T)} = \int_{\omega_z} \varphi_z \, |f - f_{\omega_z}|^2 \,\mathrm{d}x.$$

The combination of the previous estimates leads to

$$\begin{aligned}\mathbb{I} &\leqslant C_{\mathrm{PF}}(\omega_z^\star)\, \left\|\varphi_z^{1/2}(f-f_{\omega_z})\right\|_{L^2(\omega_z)} \|\nabla v\|_{L^2(\omega_z^\star)} \\ &\leqslant C_{\mathrm{PF}}(\omega_z^\star)\lambda_{\min,\omega_z^\star}^{-1/2} \left\|\varphi_z^{1/2}(f-f_{\omega_z})\right\|_{L^2(\omega_z)} \left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(\omega_z^\star)}.\end{aligned}$$

Similar to (3.20), for every $F \in \mathcal{E}^\star(\Gamma_N) \cap \mathcal{E}^\star(z)$ and $E \in \mathcal{E}(\Gamma_N)$ with $F \subseteq E$, it holds

$$\|g^\star - \sigma_h \cdot \nu_F\|_{L^2(F)}^2 \leqslant \frac{1}{(n-1)!} \left\|\varphi_z^{1/2}(g - \sigma_h \cdot \nu_E)\right\|_{L^2(E)}^2.$$

For non-Neumann sides $F \in \mathcal{E}^\star(z)$, elementary calculations show

$$|F|\,|[\sigma_h \cdot \nu_F]_F|^2 = \frac{1}{(n-1)!} \left\|\varphi_z^{1/2}[\sigma_h \cdot \nu_E]_E\right\|_{L^2(E)}^2.$$

Since $[\sigma_h \cdot \nu_F]_F = 0$ on every side $F \notin \bigcup \mathcal{E}(z)$ or $F \in \mathcal{E}^\star(\Gamma_D)$, the second term reduces to

$$\begin{aligned}\mathbb{II} &\coloneqq -\sum_{F\in\mathcal{E}^\star(z)} \int_F (v - v_{\omega_z^\star})[\sigma_h \cdot \nu_F]_F \,\mathrm{d}s = -\sum_{F\in\mathcal{E}^\star(z)} \int_{F\cap\bigcup\mathcal{E}(z)} [\sigma_h \cdot \nu_F]_F (v - v_{\omega_z^\star})\,\mathrm{d}s \\ &\leqslant \sum_{F\in\mathcal{E}^\star(z), F\subseteq\bigcup\mathcal{E}(z)\setminus\mathcal{E}(\Gamma_D)} \left(|F|^{n/(2(n-1))}\,|[\sigma_h \cdot \nu_F]_F|\right) \left(|F|^{1-n/(2(n-1))} \left| \fint_F (v - v_{\omega_z^\star})\,\mathrm{d}s\right|\right) \\ &\leqslant \left(\sum_{E\in\mathcal{F}(z)} \frac{n\,|E|^{1/(n-1)}}{(n!)^{n/(n-1)}} \left\|\varphi_z^{1/2}[\sigma_h \cdot \nu_E]_E\right\|_{L^2(E)}^2\right)^{1/2} \\ &\qquad \times \left(\sum_{F\in\mathcal{E}^\star(z), F\subseteq\bigcup\mathcal{E}(z)\setminus\mathcal{E}(\Gamma_D)} |F|^{1-1/(n-1)} \left(\fint_F (v - v_{\omega_z^\star})\,\mathrm{d}s\right)^2\right)^{1/2}.\end{aligned}$$

For any $F \in \mathcal{E}^\star(z), F \subseteq \bigcup\mathcal{E}(z)\setminus\mathcal{E}(\Gamma_D)$, the side patch $\omega_F^\star$ consists of one or two neighbouring elements $T^\star \in \mathcal{T}^\star$ with $T^\star = \mathrm{conv}\{F, \{\mathrm{mid}(T)\}\}$ where $\mathrm{mid}(T)$ is the midpoint of the element $T \in \mathcal{T}$ with $T^\star \subseteq T$. The trace identity (Lemma 2.2.17) for any $T^\star = \mathrm{conv}\{F, \{\mathrm{mid}(T)\}\}$ reads

$$\frac{1}{|F|}\int_F w\,\mathrm{d}s = \frac{1}{|T^\star|}\int_{T^\star} w\,\mathrm{d}x + \frac{1}{n\,|T^\star|}\int_{T^\star} (x - \mathrm{mid}(\mathcal{T}))\cdot\nabla w\,\mathrm{d}x.$$

The weighted summation of the two equalities leads to

$$\frac{|\omega_F^\star|}{|F|}\int_F w\,\mathrm{d}s = \int_{\omega_F^\star} w\,\mathrm{d}x + \frac{1}{n}\int_{\omega_F^\star} (x - \mathrm{mid}(\mathcal{T})(x))\cdot\nabla w\,\mathrm{d}x.$$

A Hölder inequality $X \cdot Y \leqslant |X|\,|Y|$ in $\mathbb{R}^2$ with the vectors

$$X \coloneqq \left(C_{\mathrm{PF}}(\omega_z^\star)\,|\omega_F^\star|^{-1/2}, \|\bullet - \mathrm{mid}(\mathcal{T})\|_{L^2(\omega_F^\star)}/(n\,|\omega_F^\star|)\right) \quad \text{and}$$

$$Y \coloneqq \left(\|w\|_{L^2(\omega_F^\star)} / C_{\mathrm{PF}}(\omega_z^\star), \|\nabla w\|_{L^2(\omega_F^\star)} \right)$$

yields

$$\begin{aligned}\left(\fint_F w \,\mathrm{d}s \right)^2 &\leqslant \left(\|w\|_{L^2(\omega_F^\star)} |\omega_F^\star|^{-1/2} + \frac{1}{n|\omega_F^\star|} \|\bullet - \mathrm{mid}(\mathcal{T})\|_{L^2(\omega_F^\star)} \|\nabla w\|_{L^2(\omega_F^\star)} \right)^2 \\ &\leqslant \left(C_{\mathrm{PF}}(\omega_z^\star)^2 |\omega_F^\star|^{-1} + \frac{\|\bullet - \mathrm{mid}(\mathcal{T})\|^2_{L^2(\omega_F^\star)}}{n^2 |\omega_F^\star|^2} \right) \left(\frac{\|w\|^2_{L^2(\omega_F^\star)}}{C_{\mathrm{PF}}(\omega_z^\star)^2} + \|\nabla w\|^2_{L^2(\omega_F^\star)} \right)\end{aligned}$$

The sum over all $F \in \mathcal{F}(z)$ and a Poincaré or Friedrichs inequality for $w = v - v_{\omega_z^\star}$ lead to

$$\begin{aligned}&\sum_{F\in\mathcal{F}(z)} |F|^{1-1/(n-1)} \left(\fint_F (v - v_{\omega_z^\star}) \,\mathrm{d}s \right)^2 \\ &\qquad \leqslant 2 \|\nabla v\|^2_{L^2(\omega_z^\star)} \max_{F\in\mathcal{F}(z)} |F|^{1-1/(n-1)} \left(C_{\mathrm{PF}}(\omega_z^\star)^2 |\omega_F^\star|^{-1} + \frac{\|\bullet - \mathrm{mid}(\mathcal{T})\|^2_{L^2(\omega_F^\star)}}{n^2 |\omega_F^\star|^2} \right).\end{aligned}$$

The previous estimates prove that $\mathbb{III}$ is bounded by

$$\begin{aligned}&\lambda_{\min,\omega_z^\star}^{-1/2} \left\| \mathbb{S}^{1/2} \nabla v \right\|_{L^2(\omega_z^\star)} \left(\sum_{E\in\mathcal{E}(z)} |E|^{1/(n-1)} \|[\sigma_h \cdot \nu_E]_E\|^2_{L^2(E)} \right)^{1/2} \\ &\quad \times \frac{\sqrt{2n}}{(n!)^{n/(2(n-1))}} \max_{F\in\mathcal{F}(z)} |F|^{1/2-1/(2(n-1))} \left(C_{\mathrm{PF}}(\omega_z^\star)^2 |\omega_F^\star|^{-1} + \frac{\|\bullet - \mathrm{mid}(\mathcal{T})\|^2_{L^2(\omega_F^\star)}}{n^2 |\omega_F^\star|^2} \right)^{1/2}.\end{aligned}$$

A Hölder inequality concludes the proof. □

Remark 3.4.3. *The analysis in Carstensen and Merdon (2013+) also handles more general situations with L^p norms and it recovers the explicit bounds from Veeser and Verfürth (2009) with even better constants in some benchmark examples. A further comparison is possible with Carstensen and Funken (1999) where $c_1(z)$ and $c_2(z)$ are found through the numerical solve of local analytic eigenvalue problems.*

3.4.2 Efficiency by Bubble Technique

The proof of the efficiency of η_{R} or $\eta_{\mathrm{R}}^\star$ relies on the bubble technique of Verfürth (1996). The bubble functions read

$$b_T \coloneqq \prod_{z\in\mathcal{N}(T)} \varphi_z \in \mathcal{P}_{n+1}(T) \quad \text{for } T \in \mathcal{T} \qquad \text{and} \qquad b_E \coloneqq \prod_{z\in\mathcal{N}(E)} \varphi_z \in \mathcal{P}_n(\omega_E) \quad \text{for } E \in \mathcal{E}.$$

Lemma 3.4.4 (Local efficiency (Verfürth, 1994))**.** *For discrete functions $w_T \in \mathcal{P}_0(T)$ and $w_E \in \mathcal{P}_0(E)$, it holds*

(a) $c_1 \|w_T\|_{L^2(T)} \leqslant \sup_{v\in\mathcal{P}_k(T)} \int_T w_T b_T v \,\mathrm{d}x / \|v\|_{L^2(T)} \leqslant \|w_T\|_{L^2(T)}$,
(b) $c_2 \|w_E\|_{L^2(E)} \leqslant \sup_{v\in\mathcal{P}_k(E)} \int_E w_E b_E v \,\mathrm{d}s / \|v\|_{L^2(E)} \leqslant \|w_E\|_{L^2(E)}$,
(c) $c_3 h_T^{-1} \|b_T w_T\|_{L^2(T)} \leqslant \|\nabla(b_T w_T)\|_{L^2(T)} \leqslant c_4 h_T^{-1} \|b_T w_T\|_{L^2(T)}$ *for* $T \in \mathcal{T}(E)$,
(d) $c_5 h_T^{-1} \|b_E w_E\|_{L^2(T)} \leqslant \|\nabla(b_E w_E)\|_{L^2(T)} \leqslant c_6 h_T^{-1} \|b_E w_E\|_{L^2(T)}$ *for* $T \in \mathcal{T}(E)$,
(e) $\|b_E w_E\|_{L^2(T)} \leqslant c_7 h_T^{1/2} \|w_E\|_{L^2(E)}$ *for* $T \in \mathcal{T}(E)$.

The constants $c_1, \ldots, c_7$ *depend on the shape of* T *and* E *but not on the diameter* h_T.

Proof. A proof can be found in Verfürth (1994, Lemma 5.1 on page 455). □

Theorem 3.4.5 (Efficiency of η_R)**.** *For* $e \in V$ *with* $\mathrm{Res}(v) = a(e,v)$ *for all* $v \in V$, *it holds*

(a) $h_T \lambda_{\max,T}^{-1/2} \|f\|_{L^2(T)} \lesssim \left\|\mathbb{S}^{1/2}\nabla e\right\|_{L^2(T)} + \mathrm{osc}(\lambda_{\max,\mathcal{T}}^{-1/2} f, T)$ *for* $T \in \mathcal{T}$,
(b) $h_E^{1/2} \lambda_{\max,\omega_E}^{-1/2} \|[\sigma_h \cdot \nu_E]_E\|_{L^2(E)} \lesssim \left\|\mathbb{S}^{1/2}\nabla e\right\|_{L^2(\omega_E)} + \mathrm{osc}(\lambda_{\max,\omega_E}^{-1/2} f, \mathcal{T}(E))$ *for* $E \in \mathcal{E}(\Omega)$,
(c) $h_E^{1/2} \lambda_{\max,\omega_E}^{-1/2} \|g - \sigma_h \cdot \nu_E\|_{L^2(E)} \lesssim \left\|\mathbb{S}^{1/2}\nabla e\right\|_{L^2(\omega_E)} + \mathrm{osc}(\lambda_{\max,\mathcal{T}}^{-1/2} g, \omega_E)$
$\qquad + \mathrm{osc}(\lambda_{\max,\omega_E}^{-1/2} f, \mathcal{T}(E))$ *for* $E \in \mathcal{E}(\Gamma_N)$,
(d) $\eta_R \lesssim |||e|||$.

Proof. A triangle inequality shows

$$h_T \|f\|_{L^2(T)} \leqslant h_T \|f_T\|_{L^2(T)} + \mathrm{osc}(f, T).$$

Lemma 3.4.4.(a) and $\int_T \sigma_h \cdot \nabla(b_T v_T) \,\mathrm{d}x = 0$ yield

$$\begin{aligned} h_E \|f_T\|_{L^2(T)} &\leqslant \frac{h_T}{c_1} \sup_{v_T\in\mathcal{P}_0(T)} \int_T f_T b_T v_T \,\mathrm{d}x / \|v_T\|_{L^2(T)} \\ &\leqslant \frac{h_T}{c_1} \sup_{v_T\in\mathcal{P}_0(T)} \mathrm{Res}(b_T v_T) / \|v_T\|_{L^2(T)} - \int_T (f - f_T) b_T v_T \,\mathrm{d}x / \|v_T\|_{L^2(T)} \end{aligned}$$

The relation $\mathrm{Res}(b_T v_T) = a(e, b_T v_T)$, a Hölder inequality, Lemma 3.4.4.(c) and $b_T \leqslant 1$ result in

$$h_T \|f_T\|_{L^2(T)} \leqslant \frac{c_4}{c_1} \left(\left\|\mathbb{S}^{1/2}\nabla e\right\|_{L^2(T)} + \mathrm{osc}(f, \mathcal{T}) \right).$$

This concludes the proof of (a). Lemma 3.4.4.(b) and the integration by parts $\int_{\omega_E} \sigma_h \cdot \nabla(b_E v_E) \,\mathrm{d}x = \int_E [\sigma_h \cdot \nu_E]_E b_E v_E \,\mathrm{d}s$ yield

$$\begin{aligned} h_T^{1/2} \|[\sigma_h \cdot \nu_E]_E\|_{L^2(E)} &\leqslant \frac{h_T^{1/2}}{c_2} \sup_{v_E\in\mathcal{P}_0(E)} \int_E [\sigma_h \cdot \nu_E]_E b_E v_E \,\mathrm{d}s / \|v_E\|_{L^2(E)} \\ &= \frac{h_T^{1/2}}{c_2} \sup_{v_E\in\mathcal{P}_0(E)} \left(\mathrm{Res}(v_E b_E) - \int_{\omega_E} f v_E b_E \,\mathrm{d}x \right) / \|v_E\|_{L^2(E)}. \end{aligned}$$

A Hölder inequality, Lemma 3.4.4.(d), Lemma 3.4.4.(e) and $b_E \leqslant 1$ yield

$$h_T^{1/2} \left\| [\sigma_h \cdot \nu_E]_E \right\|_{L^2(E)} \leqslant \frac{c_6}{c_2} \left\| \mathbb{S}^{1/2} \nabla e \right\|_{L^2(\omega_E)} + \frac{c_7 h_T}{c_2} \left\| f \right\|_{L^2(\omega_E)}.$$

Assertion (a) for the second term on the right-hand side and the equivalence $h_E \approx h_T$ (by shape regularity, see Definition 2.3.6) conclude the proof. In case of Neumann sides $E \in \mathcal{E}(\Gamma_N)$ in assertion (c), the average $g_E := \fint_E g \, \mathrm{d}s$ enters by a triangle inequality

$$h_E^{1/2} \left\| g - \sigma_h \cdot \nu_E \right\|_{L^2(E)} \leqslant h_E^{1/2} \left\| g_E - \sigma_h \cdot \nu_E \right\|_{L^2(E)} + \operatorname{osc}(g, E).$$

The rest of the proof is very similar to the proof of (b). The last assertion (d) is a combination of (a)-(c) together with some finite overlap arguments. The multiplication with $\lambda_{\max,T}^{-1/2}$ or $\lambda_{\max,\omega_E}^{-1/2}$ on both sides and $\left\| \mathbb{S}^{1/2} \nabla e \right\|_{L^2(T)} \leqslant \lambda_{\max,T}^{1/2} \left\| \nabla e \right\|_{L^2(T)}$ lead to the weighted estimates. □

Remark 3.4.6. *Opposite to Definition 3.4.1, $\lambda_{\min,\omega_z^\star}^{-1/2}$ enters in the local error estimator contributions of $\eta_{\mathrm{R}}^\star$ for optimal reliability constants. The proof of efficiency for $\eta_{\mathrm{R}}^\star$ by Theorem 3.4.5 requires only little modifications, e.g., the factor $\lambda_{\max,T}^{1/2} \lambda_{\max,\omega_z^\star}^{-1/2}$ might enter the local efficiency constants.*

4 Error Analysis for the Poisson Model Problem

The Poisson model problem describes various kinds of physical phenomena that can be expressed with potentials, e.g. electrostatics, gravitation and hydrodynamics. This chapter deals with the conforming FEM in Section 4.2 and the nonconforming Crouzeix-Raviart FEM in Section 4.4 and the Raviart-Thomas mixed FEM in Section 4.7. In all cases, the error analysis enables guaranteed upper bounds for the (broken) energy norm with the error estimators of Chapter 3 also for mixed inhomogeneous boundary conditions.

4.1 Setting

The generalised interface problem for the Laplacian with data $f \in L^2(\Omega)$, $g \in L^2(\Gamma_N)$, $u_D \in L^2(\Gamma_D)$ and $\mathbb{S}$ from Subsection 2.3.1 seeks $u \in u_D + V$ with

$$-\operatorname{div}(\mathbb{S}\nabla u) = f \text{ in } \Omega \quad \text{and} \quad q \cdot \nu = g \text{ along } \Gamma_N. \tag{4.1}$$

The stress tensor $\sigma \coloneqq \mathbb{S}\nabla u$ satisfies the equilibrium equation $\operatorname{div}\sigma + f = 0$. The weak formulation results in the variational equality

$$a(u, v) = F(v) \qquad \text{for all } v \in V. \tag{4.2}$$

With this, the solution u is not only the best-approximation in the energy norm, but also the minimiser of the energy

$$E(v) \coloneqq a(v, v)/2 - F(v) \quad \text{amongst } v \in u_D + V. \tag{4.3}$$

In the sequel the diffusion tensor is assumed to be piecewise constant, i.e., $\mathbb{S} \in \mathcal{P}_0(\mathcal{T}; \mathbb{R}^{n\times n})$ with piecewise smallest and largest eigenvalues $\lambda_{\min,\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$ and $\lambda_{\max,\mathcal{T}} \in \mathcal{P}_0(\mathcal{T})$ from (3.2).

4.2 Error Analysis for Conforming $\mathcal{P}_1$-FEM

Recall the ansatz space $V(\mathcal{T}) = \mathcal{P}_1(\mathcal{T}) \cap V$ for the $\mathcal{P}_1$ conforming finite element method. The conforming finite element method approximates the Dirichlet data u_D by its nodal interpolation $u_{D,h} \coloneqq \sum_{z\in\mathcal{N}(\Gamma_D)} u_D(z)\varphi_z$ and seeks $u_h \in u_{D,h} + V(\mathcal{T})$ with

$$a(u_h, v_h) = F(v_h) \qquad \text{for all } v_h \in V(\mathcal{T}). \tag{4.4}$$

The discrete flux $\sigma_h \coloneqq \mathbb{S}\nabla u_h$ leads to the residual $\mathrm{Res} \in V^\star$ in the form (3.1), i.e.,

$$\mathrm{Res}(v) \coloneqq a(u-u_h, v) = \int_\Omega fv \,\mathrm{d}x - \int_\Omega \sigma_h \cdot \nabla v \,\mathrm{d}x \qquad \text{for } v \in V(\mathcal{T}).$$

By (4.4), the residual Res satisfies the Galerkin orthogonality

$$\mathrm{Res}(\varphi_z) = 0 \quad \text{for all free nodes } z \in \mathcal{M} = \mathcal{N}\backslash\mathcal{N}(\Gamma_D).$$

4.2.1 Error Decomposition

In case of homogeneous Dirichlet boundary data, Section 3.1 already states the identity $|||\mathrm{Res}|||_\star = |||u-u_h|||$. In the case of inhomogeneous Dirichlet boundary data there is an additional error that needs to be handled separately.

Lemma 4.2.1 (Dirichlet Error Split-Off)**.** *For $e \coloneqq u - u_h$ and any $w_D \in H^1(\Omega)$ with $w_D = u_D - u_{D,h}$ along Γ_D and $\mathbb{S}\nabla w_D \cdot \nu = 0$ along Γ_N, it holds*

$$|||e|||^2 \leqslant |||\mathrm{Res}|||_\star^2 + |||w_D|||^2 .$$

Proof. Let $w \in H^1(\Omega)$ be the harmonic extension with $w = u_D - u_{D,h}$ along Γ_D, $\mathrm{div}(\mathbb{S}\nabla w) = 0$ and $\mathbb{S}\nabla w \cdot \nu = 0$ along Γ_N. Then, for any w_D that complies with the requirements, $w_D - w \in V$ and an integration by parts show

$$\begin{aligned}&\int_\Omega \mathbb{S}\nabla w \cdot \nabla(w_D - w)\,\mathrm{d}x \\ &\quad = \int_\Omega \mathrm{div}(\mathbb{S}\nabla w)(w_D - w)\,\mathrm{d}x + \int_{\Gamma_N} \mathbb{S}w \cdot \nu (w_D - w)\,\mathrm{d}s + \int_{\Gamma_D} \mathbb{S}w \cdot \nu (w_D - w)\,\mathrm{d}s = 0.\end{aligned}$$

This proves the Pythagoras theorem

$$|||w|||^2 = |||w_D|||^2 - |||w_D - w|||^2 \leqslant |||w_D|||^2 . \tag{4.5}$$

For the special choice $w_D = e$, it follows

$$|||e|||^2 = |||e-w|||^2 + |||w|||^2 . \tag{4.6}$$

Another integration by parts shows $\int_\Omega \mathbb{S}\nabla w \cdot \nabla v \,\mathrm{d}x = 0$ and thus

$$\mathrm{Res}(v) = a(e-w, v) \qquad \text{for all } v \in V.$$

Hence, $|||\mathrm{Res}|||_\star \leqslant |||e-w|||$. Since $v = e - w \in V$ is a valid test function, also the converse holds, i.e.,

$$|||\mathrm{Res}|||_\star = |||e-w||| .$$

The combination with (4.5) and (4.6) concludes the proof. □

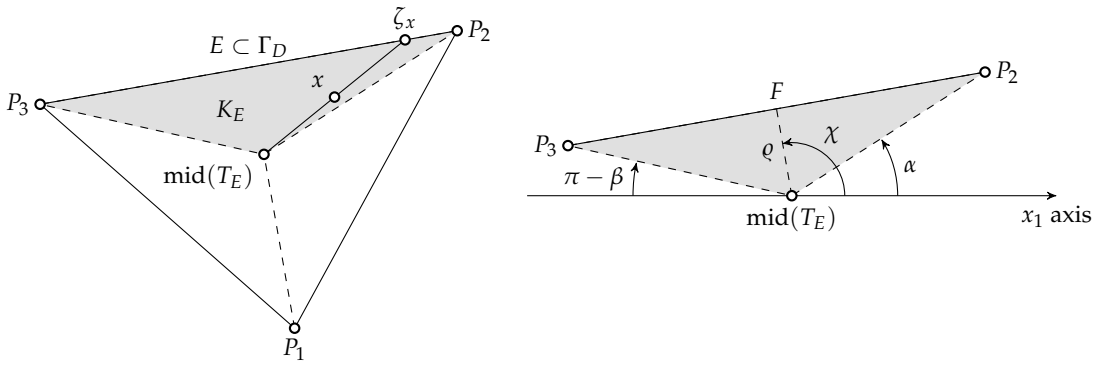

Figure 4.1: Notation for the design of the boundary extension w_D.

4.2.2 Boundary Extension

This subsection offers an explicit design of some boundary extension $w_D \in H^1(\Omega)$ with the properties of Lemma 4.2.1 and an upper bound for $|||w_D|||$ to allow guaranteed error control also in case of inhomogeneous Dirichlet boundary data. To apply this design in the scope of Lemma 4.2.1, set $v_D = (u_D - u_{D,h})|_{\Gamma_D}$ below.

Theorem 4.2.2. *Assume that $v_D \in H^1(\Gamma_D) \cap C(\Gamma_D)$ satisfies $v_D \in H^1_0(E) \cap H^2(E)$ for all $E \in \mathcal{E}(\Gamma_D)$ and let $\partial^2_{\mathcal{E}} v_D/\partial s^2$ denote the edgewise second surface derivative of v_D along Γ_D. Then there exists $w_D \in H^1(\Omega)$ with $w_D|_{\Gamma_D} = v_D$ and*

$$|||w_D||| \leqslant C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) \left\| h_{\mathcal{E}}^{3/2} \lambda_{\max,\mathcal{T}}^{1/2} \partial^2_{\mathcal{E}} v_D/\partial s^2 \right\|_{L^2(\Gamma_D)}$$
$$\|w_D\|_{L^2(\Omega)} \leqslant C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D)) \left\| h_E^{5/2} \partial^2_{\mathcal{E}} v_D/\partial s^2 \right\|_{L^2(E)}.$$

The constants $C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) \lesssim 1$ and $C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D)) \lesssim 1$ only depend on the shape of the triangle but not on the mesh size. In the 2D case, $C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D))$ and $C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D))$ are bounded by

$$C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) \coloneqq \max_{E\in\mathcal{E}(\Gamma_D)} \frac{\sqrt{(\pi\delta + h_E)^2 + h_E^2}}{\pi^2\sqrt{2h_E\varrho}} \quad \text{and} \quad C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D)) \coloneqq \max_{E\in\mathcal{E}(\Gamma_D)} \frac{\delta}{2\pi^2\sqrt{h_E\varrho}}$$

where $\delta \coloneqq \max_{x\in E} |x - \mathrm{mid}(T_E)|$ and $\varrho \coloneqq \mathrm{dist}(\mathrm{mid}(T_E), E)$ of the adjacent triangle T_E of $E \in \mathcal{E}(\Gamma_D)$.

Remark 4.2.3. *For right isosceles triangles, as in the first three triangulations of Figure 4.2, the constants equal $C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) = 0.4980$ and $C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D)) = 0.0654$. For the octagon domain in Figure 4.2, the constants equals $C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) = 0.3589$ and $C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D)) = 0.0513$.*

Proof. The proof employs an explicit construction of w_D from Bartels et al. (2004) and is repeated here to calculate $C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D))$ and $C_{\mathrm{D},2}(\mathcal{E}(\Gamma_D))$ for guaranteed error control for $n = 2$. The case $n = 3$ allows similar arguments. Consider a triangle $T_E = \mathrm{conv}\{P_1, P_2, P_3\} \in \mathcal{T}$ with a Dirichlet side $E \coloneqq \mathrm{conv}\{P_2, P_3\} \in \mathcal{E}(\Gamma_D)$ as in Figure 4.1. The connection between the center of gravity $\mathrm{mid}(T)$ and the three vertices of T_E results in

three subtriangles of the same area. Let K_E denote the subtriangle of Figure 4.1 with $E = \partial K_E \cap \partial T$. For every point $x \in K_E \backslash \{\mathrm{mid}(T_E)\}$, there exist some unique $\zeta_x \in E$ and $0 < \theta_x \leqslant 1$ with $x = (1-\theta_x)\,\mathrm{mid}(T_E) + \theta_x \zeta_x$. Then,

$$w_D(x)|_{K_E} \coloneqq \begin{cases} \theta_x v_D(\zeta_x) & \text{for } x \in K_E \backslash \{\mathrm{mid}(T_E)\}, \\ 0 & \text{else.} \end{cases}$$

On $T_E \backslash K_E$ and every element $T \in \mathcal{T}$ with $|\partial T \cap \Gamma_D| = 0$, w_D is set to zero.

Without loss of generality, we assume $\mathrm{mid}(T_E) = 0 \in \mathbb{R}^2$. Polar coordinates yield the parameterisation

$$K_E = \{x = (r\cos(\varphi), r\sin(\varphi)) \,\big|\, \alpha < \varphi < \beta,\ 0 < r < R(\varphi) \coloneqq |\zeta_x|\}$$

where α and β are the angles at the points P_2 and P_3, respectively, as depicted in the right part of Figure 4.1. For $x \in K_E$ and $\zeta_x = \varrho \nu_E + s\tau_E$ with normal vector ν_E, tangential vector τ_E and height $\varrho = \mathrm{dist}(\mathrm{mid}(T_E), E)$, it holds $R(\varphi)^2 = |\zeta_x|^2 = \varrho^2 + s^2(\varphi)$. Furthermore, it holds $w_D(x) = v_D(\varphi) r / R(\varphi)$ and

$$\begin{aligned} |\nabla w_D(r,\varphi)|^2 &= |\partial w_D/\partial r|^2 + |\partial w_D/\partial \varphi|^2 / r^2 \\ &= |v_D(\varphi)/R(\varphi)|^2 + |R(\varphi) v_D'(\varphi) - R'(\varphi) v_D(\varphi)|^2 / R^2(\varphi). \end{aligned}$$

The introduction of the angle χ of the perpendicular point F depicted in Figure 4.1 allows for the expressions

$$R(\varphi) = \varrho / \cos(\varphi - \chi) \quad \text{and} \quad s(\varphi) = \varrho \tan(\varphi - \chi).$$

Elementary calculations lead to the differentials

$$R'(\varphi) = s(\varphi) R(\varphi)/\varrho, \qquad \frac{\partial s}{\partial \varphi} = \varrho / \cos^2(\varphi - \chi) = R(\varphi)^2/\varrho \quad \text{and}$$

$$v_D'(\varphi) = \frac{\partial v_D}{\partial s}\frac{\partial s}{\partial \varphi} = \frac{\partial v_D}{\partial s} R(\varphi)^2/\varrho.$$

Hence,

$$\begin{aligned} \int_{K_E} |\nabla w_D(x)|^2 \,\mathrm{d}x &= \int_\alpha^\beta \int_0^{R(\varphi)} |\nabla w_D(r,\varphi)|^2 \, r \,\mathrm{d}r \,\mathrm{d}\varphi \\ &= \frac{1}{2}\int_\alpha^\beta v_D(\varphi)^2 + \left(\frac{R^2 v_D'(\varphi) - v_D(\varphi) s}{\varrho}\right)^2 \mathrm{d}\varphi. \end{aligned}$$

The transformation $\mathrm{d}\varphi = \cos^2(\varphi - \chi)\,\mathrm{d}s/\varrho = \varrho\,\mathrm{d}s/(\varrho^2 + s^2)$ yields

$$\int_\alpha^\beta v_D(\varphi)^2 \,\mathrm{d}\varphi = \int_a^b \frac{v_D(s)^2 \varrho}{\varrho^2 + s^2}\,\mathrm{d}s \leqslant \frac{1}{\varrho}\,\|v_D\|^2_{L^2(E)}\,.$$

The same transformation and the Young inequality for some $\gamma > 0$ in the second summand show

$$\begin{aligned}\int_\alpha^\beta \left(\frac{R^2(\varphi)v_D'(\varphi) - v_D(\varphi)s}{\varrho}\right)^2 \mathrm{d}\varphi \\ &= \int_a^b \left(\frac{R^2(s)(\partial v_D/\partial s) - v_D(s)s}{\varrho}\right)^2 \frac{\varrho}{\varrho^2+s^2}\,\mathrm{d}s \\ &\leqslant \int_a^b (1+\gamma)\frac{R^4(s)}{\varrho(\varrho^2+s^2)}\left(\frac{\partial v_D}{\partial s}\right)^2 + (1+1/\gamma)\frac{v_D(s)^2 s^2}{\varrho(\varrho^2+s^2)}\,\mathrm{d}s.\end{aligned}$$

The identities $R^2(s) = \varrho^2 + s^2$ and $\delta = \max_{\varphi\in[\alpha,\beta]} R(\varphi)$ in the first term as well as the estimate $s^2/(\varrho(\varrho^2+s^2)) \leqslant 1/\varrho$ in the second term result in

$$\int_\alpha^\beta \left(\frac{R(\varphi)^2 v_D'(\varphi) - v_D(\varphi)s}{\varrho}\right)^2 \mathrm{d}\varphi \leqslant \frac{(1+\gamma)\delta^2}{\varrho}\left\|\partial v_D/\partial s\right\|_{L^2(E)}^2 + \frac{1+1/\gamma}{\varrho}\left\|v_D\right\|_{L^2(E)}^2.$$

A Friedrichs inequality for $v_D \in H_0^1(E)$ and a Poincaré inequality for $\partial v_D/\partial s$ yield

$$\begin{aligned}\int_{K_E} |\nabla w_D(x)|^2\,\mathrm{d}x &\leqslant \frac{(1+\gamma)\delta^2}{2\varrho}\left\|\partial v_D/\partial s\right\|_{L^2(E)}^2 + \frac{2+1/\gamma}{2\varrho}\left\|v_D\right\|_{L^2(E)}^2 \\ &\leqslant \left(\frac{(1+\gamma)\delta^2 h_E^2}{2\pi^2\varrho} + \frac{(2+1/\gamma)h_E^4}{2\pi^4\varrho}\right)\left\|\partial^2 v_D/\partial s^2\right\|_{L^2(E)}^2 \\ &= \left(\frac{(1+\gamma)\delta^2}{2\pi^2 h_E\varrho} + \frac{(2+1/\gamma)h_E}{2\pi^4\varrho}\right)\left\|h_E^{3/2}\partial^2 v_D/\partial s^2\right\|_{L^2(E)}^2.\end{aligned}$$

Elementary computations lead to the optimal $\gamma = h_E/(\pi\delta)$, thus

$$\left\|\mathrm{S}^{1/2}\nabla w_D\right\|_{L^2(T_E)}^2 \leqslant \lambda_{\max,T_E}\left\|\nabla w_D\right\|_{L^2(T_E)}^2 \leqslant \frac{(\pi\delta+h_E)^2 + h_E^2}{2\pi^4 h_E\varrho}\left\|h_E^{3/2}\lambda_{\max,T_E}^{1/2}\partial^2 v_D/\partial s^2\right\|_{L^2(E)}^2.$$

The same arguments show

$$\begin{aligned}\left\|w_D\right\|_{L^2(T_E)}^2 &= \int_\alpha^\beta \int_0^{R(\varphi)} w_D(r,\varphi)^2 r\,\mathrm{d}r\,\mathrm{d}\varphi = \frac{1}{4}\int_\alpha^\beta v_D(\varphi)^2 R(\varphi)^2\,\mathrm{d}\varphi \\ &\leqslant \frac{\delta^2}{4\varrho}\left\|v_D\right\|_{L^2(E)}^2 \leqslant \frac{\delta^2}{4\pi^4 h_E\varrho}\left\|h_E^{5/2}\partial_{\mathcal{E}}^2 v_D/\partial s^2\right\|_{L^2(E)}^2.\end{aligned}$$

A sum over all $E \in \mathcal{E}(\Gamma_D)$ concludes the proof. □

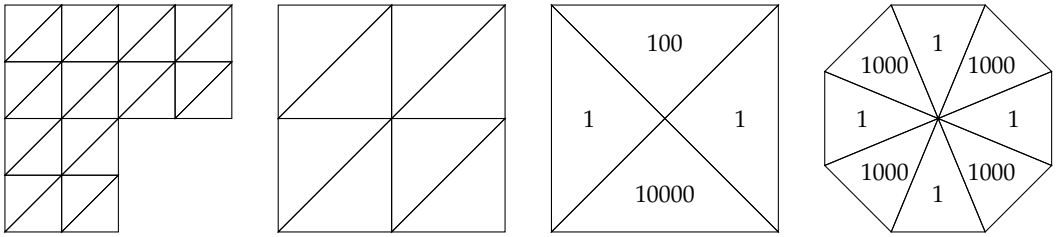

Figure 4.2: Initial triangulations for the benchmark problems of Section 4.3; from left to right the L-shaped domain, the square domain, and the square and octagon domain with discontinuous coefficients are displayed. The numbers correspond to the local diffusion coefficients for the problems where $\mathbb{S} = \alpha\mathbb{I}$ with $\alpha \not\equiv 1$.

4.3 Numerical Examples for Conforming $\mathcal{P}_1$-FEM

This section presents some numerical examples to compare the efficiency of all the error estimators η of Chapter 3 in

$$|||u - u_h|||^2 \leqslant \eta^2 + C_{\mathrm{D},1}(\mathcal{E}(\Gamma_D))^2 \left\| h_{\mathcal{E}}^{3/2} \lambda_{\max,\mathcal{T}} \partial_{\mathcal{E}}^2 u_D/\partial s^2 \right\|^2_{L^2(\Gamma_D)}$$

for the estimation of $|||u - u_h|||$ via Lemma 4.2.1. The adaptive mesh refinement in all examples is driven by the Dörfler marking of Subsubsection 2.3.4.2 with bulk parameter $\theta = 0.5$ and the elementwise refinement indicators

$$\eta(T)^2 := \frac{|T|}{\lambda_{\max,T}} \|f\|^2_{L^2(T)} + |T|^{1/2} \sum_{E\in\mathcal{E}(T)} \lambda^{-1}_{\max,\omega_E} \|[\sigma_h \cdot \nu_E]\|^2_{L^2(E)} + 0.248\, \lambda_{\max,T} \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D/\partial s^2 \right\|^2_{L^2(\partial T\cap\Gamma_D)}. \quad (4.7)$$

4.3.1 L-Shaped Domain with Constant Right-Hand Side

The first example concerns the L-shaped domain $\Omega = (-1,1)^2\backslash([0,1]\times[-1,0])$ and right-hand side $f \equiv 1$. This example employs homogeneous Dirichlet boundary conditions $u_D = 0$ along $\Gamma_D = \partial\Omega$. Since the exact solution is unknown, the energy error is calculated by Lemma 2.3.1.(b) with $|||u|||^2 = 0.214075802680976$. This reference norm was calculated with higher-order finite element methods on adaptive meshes.

Figure 4.3 displays the convergence history of the energy error $|||u - u_h|||$ for adaptive and uniform mesh refinement with respect to the number of degrees of freedom $|\mathcal{M}|$. The adaptive mesh refinements leads to the optimal convergence rate of $1/2$ with respect to the number of degrees of freedom whereas the uniform mesh refinement leads to a slower convergence of about $1/3$. The adaptive mesh on level 4 shows a finer refinement at the reentrant corner.

Figures 4.4 and 4.5 depict the efficiency indices of all tested error estimators. All indices of the equilibration error estimators are below 1.6 for uniform mesh refinement and below 1.35 for adaptive mesh refinement. The postprocessed equilibration error estimators $\eta_{\mathrm{Br}(1)}$ and $\eta_{\mathrm{LW}(1)}$ lead to significant improvements: the efficiency indices decrease to

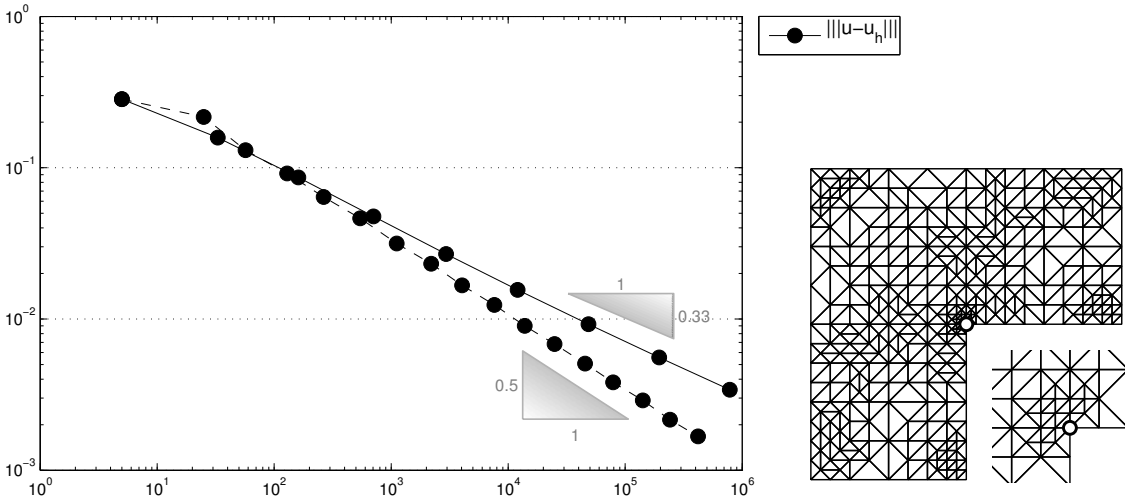

Figure 4.3: Convergence history for the energy error $|||u - u_h|||$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.3.1 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 4 and the neighbourhood of the singular point $(0, 0)$ magnified by a factor 2.

below 1.15 in case of adaptive mesh refinement. The postprocessing with two red-refinements $\eta_{\mathrm{Brr}(3)}$ even leads to efficiency indices around 1.03 which means only three percent overestimation. Hence, the postprocessing $\eta_{\mathrm{Brr}(3)}$ reduces the overestimation by a factor 10 compared to η_{B}.

4.3.2 Square with Large Oscillations

The second benchmark example employs the exact solution

$$u(x,y) = x(x-1)y(y-1)\exp\left(-100(x-1/2)^2 - 100(y-117/1000)^2\right) \in H_0^1(\Omega)$$

on the square domain $\Omega = (0,1)^2$ with full homogeneous Dirichlet data $u_D \equiv 0$. The source term $f = -\Delta u$ matches the Laplacian of the exact solution and causes big oscillations $\mathrm{osc}(f, \mathcal{T})$ on coarse triangulations as displayed in Figure 4.6. This figure also shows the oscillation term on the dual triangulation $\mathcal{T}^\star$ that is part of the error estimator η_{LW}. It is about 30 percent smaller than the oscillations on $\mathcal{T}$. Both terms are of higher order as the convergence rate is 1, while the convergence rate of the energy error is $1/2$ for both uniform and adaptive mesh refinement. Although there is no improvement of the convergence rate, which is expected a priori for convex domains, there is a shorter pre-asymptotic phase.

Figures 4.7 and 4.8 display the efficiency indices for uniform and adaptive mesh refinement with similar results as in the first example for a large number of degrees of freedom. On coarse meshes the oscillations dominate the guaranteed upper bounds and cause large efficiency indices. As expected, the postprocessed error estimators with the coupled divergence correction $\eta_{\mathrm{Bmr}(1)}$ and $\eta_{\mathrm{LWm}(1)}$ are less affected by this and $\eta_{\mathrm{Bmrr}(3)}$ leads to efficiency indices below 1.5 even on the very coarse initial triangulation.

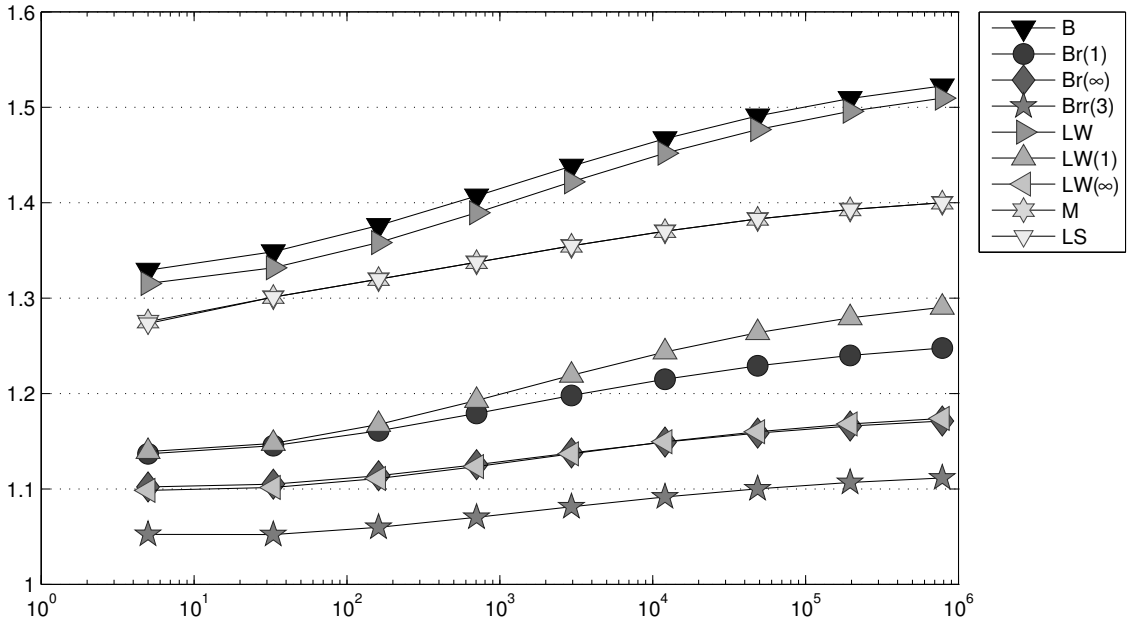

Figure 4.4: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.3.1.

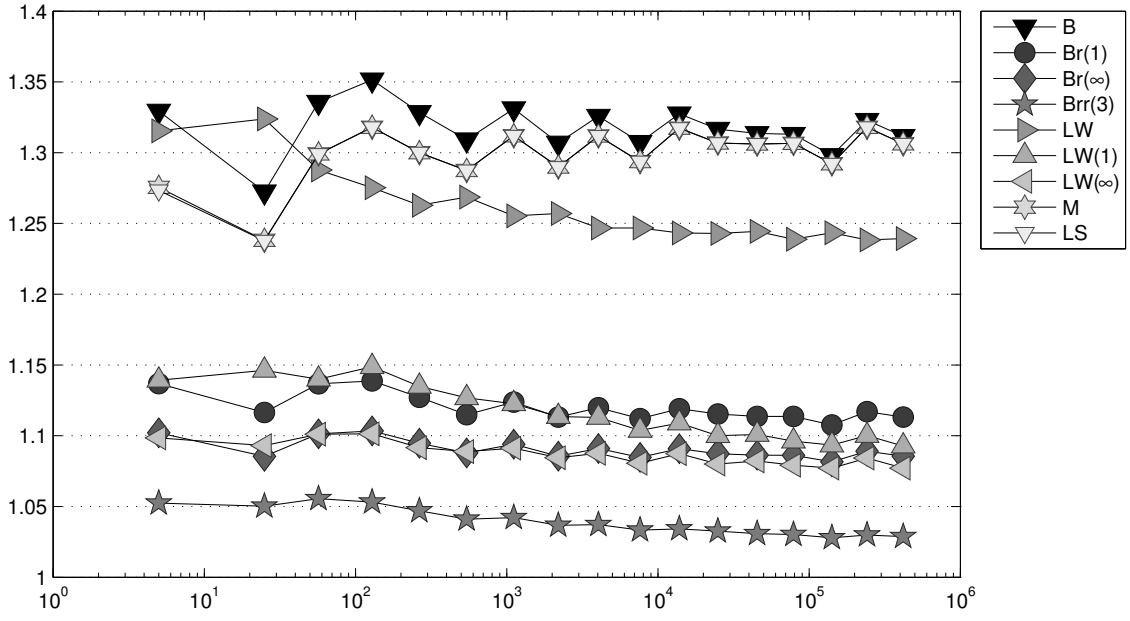

Figure 4.5: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.3.1.

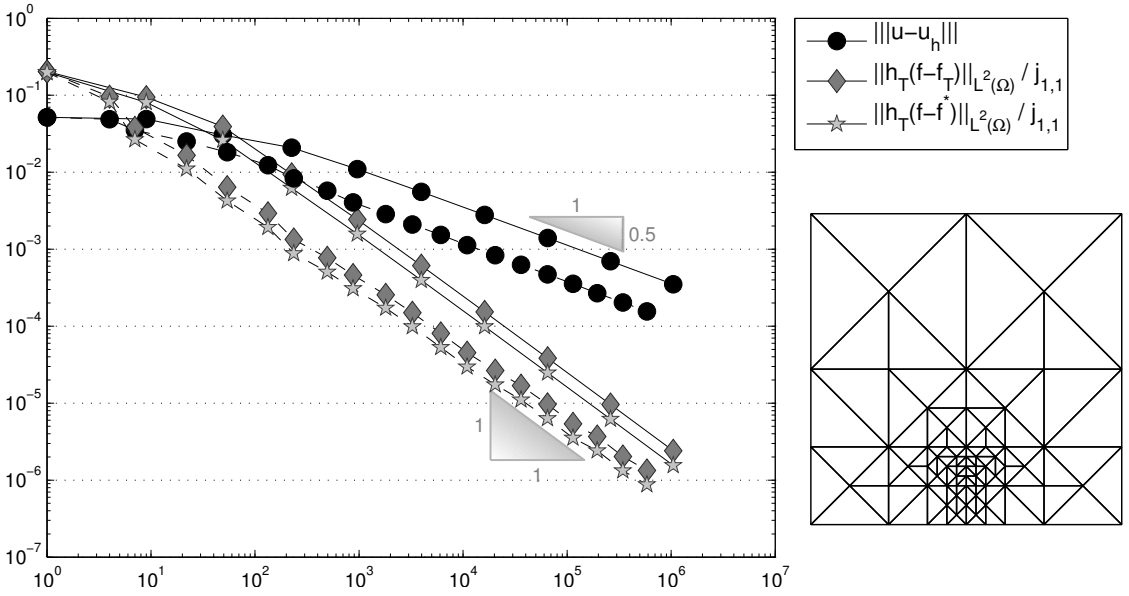

Figure 4.6: Convergence history for the energy error $|||u - u_h|||$ and the higher-order terms in the guaranteed upper bounds of η_B and η_{LW} on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.3.2 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 4.

4.3.3 Square with Discontinuous Diffusion Coefficients

The third benchmark involves $f \equiv 0$ and u_D matches the exact quadratic function $u(x,y) = (x^2 - y^2)/\alpha$ on the square domain $\Omega = (-1,1)^2$. The diffusion parameter α assumes the values $1, 100, 10000$ on subdomains depicted in Figure 4.2. The exact solution shows that α acts like a damping parameter and leads to local energy norms of smaller magnitude in the subdomains with large values of α. This behaviour is detected by the refinement indicators and reflected in the adaptive mesh depicted in Figure 4.9. Moreover, the absolute energy error on the adaptive meshes is slightly improved compared to uniform mesh refinement. The Dirichlet error contribution indeed is a term of higher order with a convergence rate of $3/4$ as expected, because $h_{\mathcal{E}}^{3/2} \approx |\mathcal{M}|^{3/4}$ for uniform mesh refinement.

Figures 4.10 and 4.11 prove that the efficiency indices are as good as in the previous examples with no jumps in the diffusion coefficient. The worse efficiency on coarse meshes is due to the extra term that measures the error for the inhomogeneous Dirichlet boundary conditions. However, this term is of higher order and so its influence diminishes on finer meshes.

4.3.4 Octagon with Discontinuous Diffusion Coefficients

The fourth example employs $f \equiv 0$ and u_D compatible with the exact solution $u(x,y) = ((cx^2 - y^2)(cy^2 - x^2))/\alpha$ with $c = \tan((3\pi)/8)^2$ on the octagon domain

$$\Omega = \text{conv}\left\{(\cos((2j+1)\pi/8), \sin((2j+1)\pi/8))\,,\ j = 0, 1, .., 7\right\}.$$

The diffusion coefficient α assumes the values 1 and 1000 as depicted in Figure 4.2.

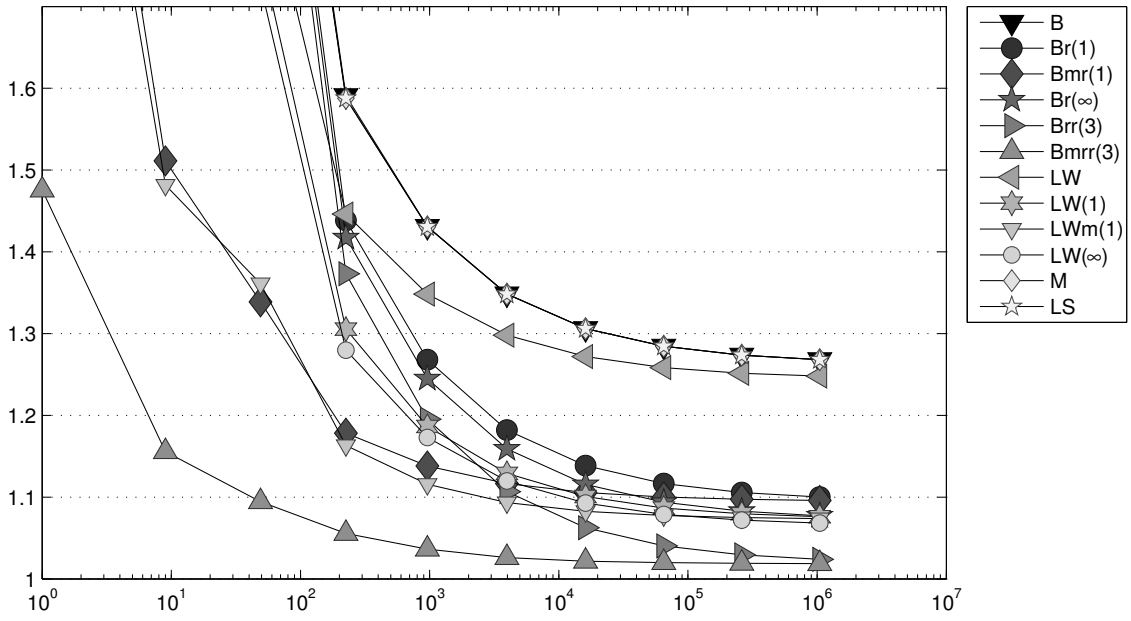

Figure 4.7: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.3.2.

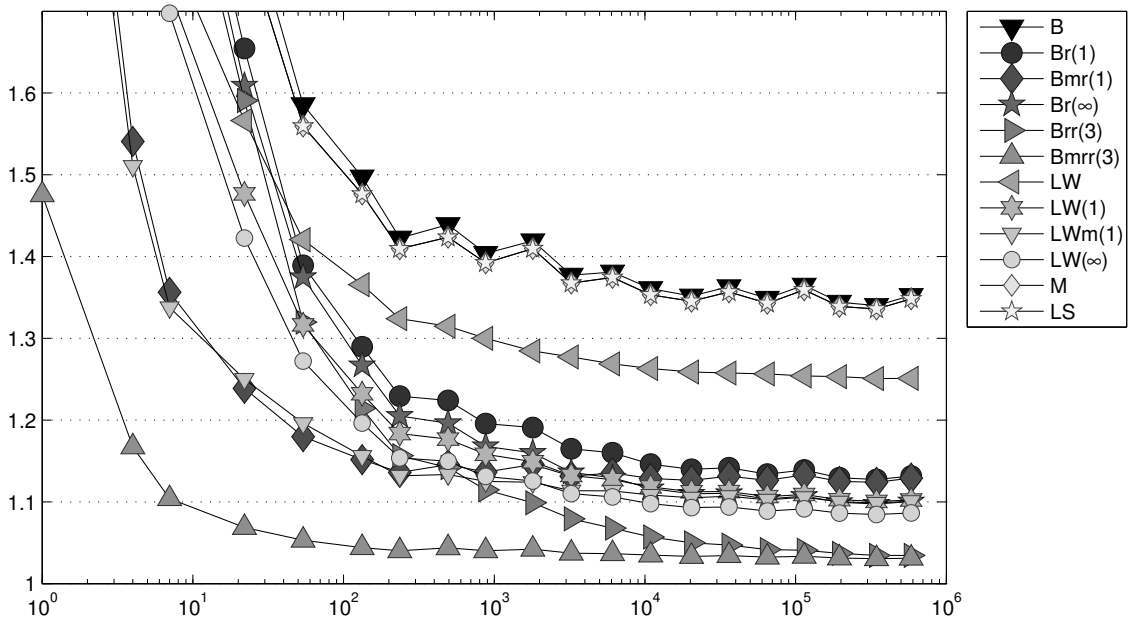

Figure 4.8: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.3.2.

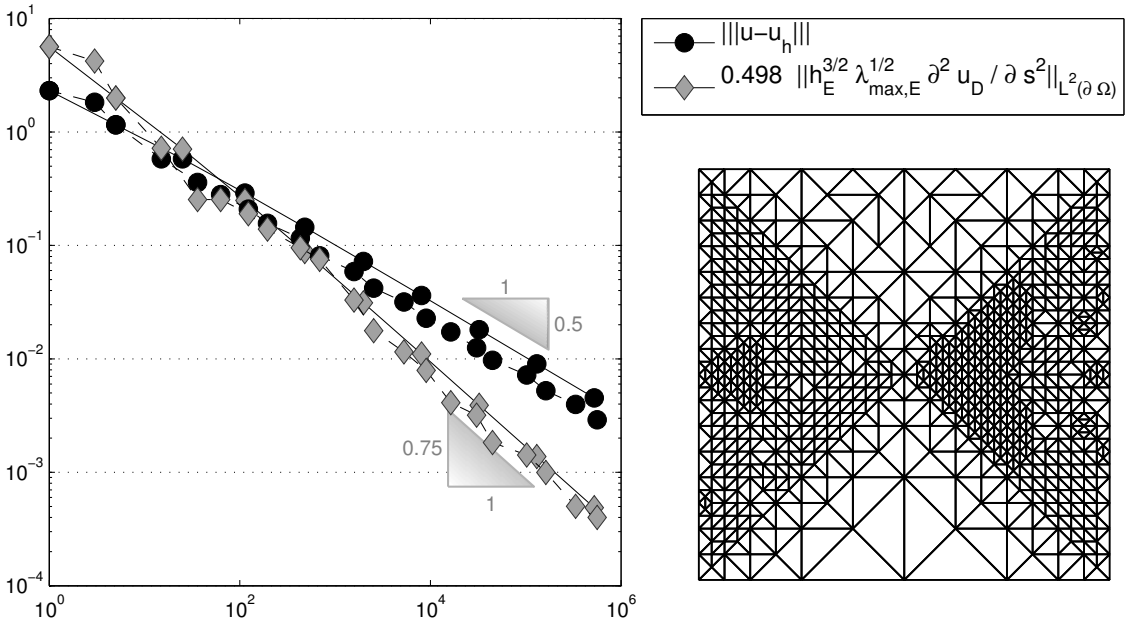

Figure 4.9: Convergence history for the energy error $|||u - u_h|||$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \lambda_{\max,\mathcal{E}}^{1/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.3.3 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 9.

Figures 4.13 and 4.14 display the efficiency indices for this example. There is no significant difference in comparison with Subsubsection 4.3.3.

4.4 Error Analysis for Nonconforming CR-FEM

The nonconforming finite element method employs the bilinear form a_{NC} from (2.6) and the nonconforming interpolation $u_{D,\mathrm{CR}} \coloneqq \sum_{E \in \mathcal{E}(\Gamma_D)} \left(\fint_E u_D \, \mathrm{d}s \right) \psi_E$ of the Dirichlet data u_D (compare with Definition 2.2.15). The Crouzeix-Raviart finite element solution $u_{\mathrm{CR}} \in u_{D,\mathrm{CR}} + \mathrm{CR}_0(\mathcal{T})$ satisfies

$$a_{\mathrm{NC}}(u_h, v_h) = F(v_h) \qquad \text{for all } v_h \in \mathrm{CR}_0(\mathcal{T}) \tag{4.8}$$

where $\mathrm{CR}_0(\mathcal{T}) = \{ v_{\mathrm{CR}} \in \mathrm{CR}(\mathcal{T}) \,|\, v_{\mathrm{CR}}(\mathrm{mid}(\mathcal{E}(\Gamma_D))) = 0 \}$. The discrete flux reads $\sigma_{\mathrm{CR}} \coloneqq \mathbb{S} \nabla_{\mathrm{NC}} u_{\mathrm{CR}}$. Subsection 4.4.1 below enables guaranteed error control in the broken energy norm $|||e|||_{\mathrm{NC}} \coloneqq a_{\mathrm{NC}}(e,e)^{1/2}$ for the error $e \coloneqq u - u_{\mathrm{CR}}$ by the dual norm control of two residuals.

4.4.1 Error Decomposition

The Helmholtz decomposition from Theorem 2.1.12 splits the piecewise gradient of the error

$$\nabla_{\mathrm{NC}} e = \nabla \alpha + \mathbb{S}^{-1} \operatorname{Curl} \beta \tag{4.9}$$

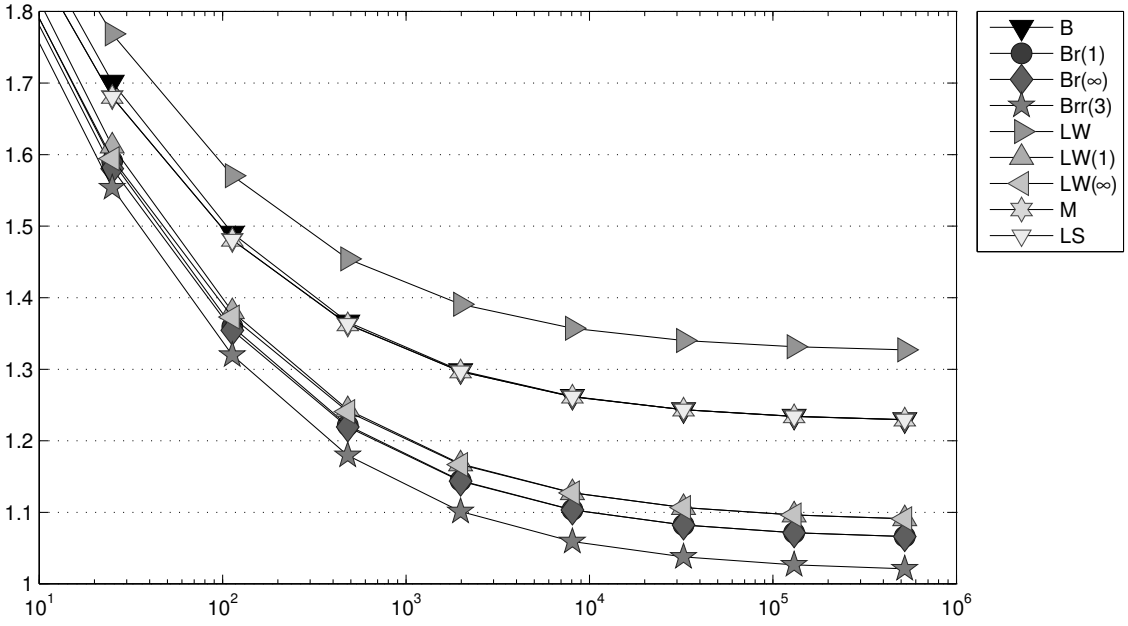

Figure 4.10: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.3.3.

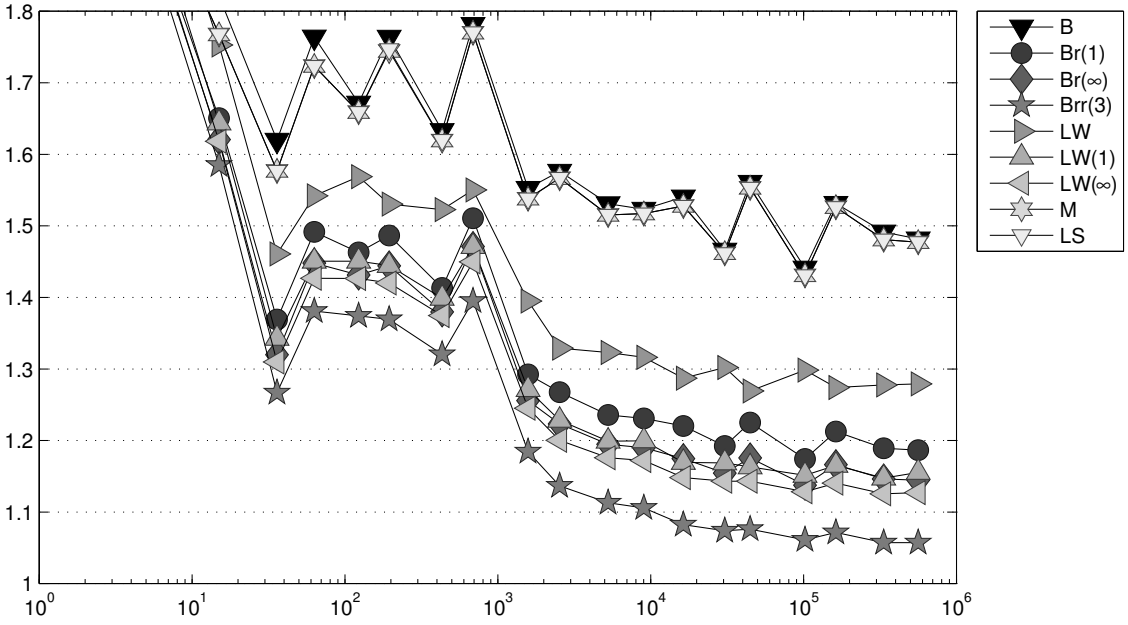

Figure 4.11: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.3.3.

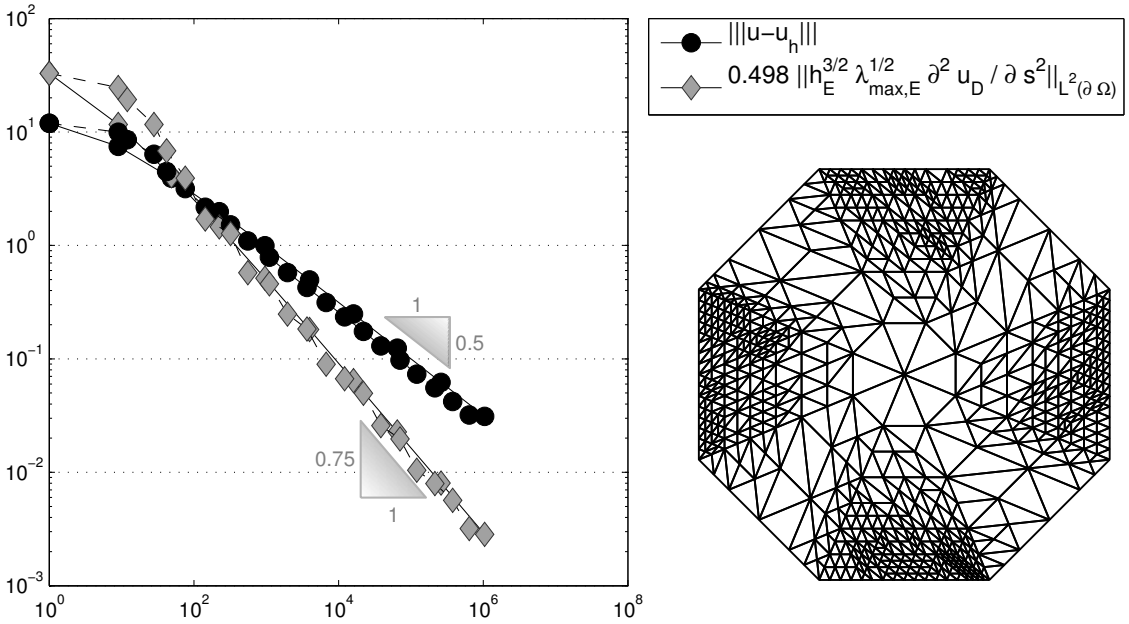

Figure 4.12: Convergence history for the energy error $|||u - u_h|||$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \lambda_{\max,\mathcal{E}}^{1/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.3.4 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 8.

into some $\alpha \in H^1_D(\Omega)$ and $\beta \in H^1(\Omega; \mathbb{R}^s)$ with $\operatorname{Curl} \beta \cdot \nu = 0$ along Γ_N. Recall that $s = 1$ for $n = 2$ and $s = 3$ for $n = 3$ as introduced in Subsection 2.1.4. The idea to employ the Helmholtz decomposition dates back to Dari et al. (1996). However, here the two error contributions are identified as dual norms of two residuals in the spirit of Carstensen (2005); Carstensen and Merdon (2013). The first one reads

$$\operatorname{Res}(v) := \int_\Omega f v \,\mathrm{d}x + \int_{\Gamma_N} g v \,\mathrm{d}s - \int_\Omega \sigma_{\mathrm{CR}} \cdot \nabla v \,\mathrm{d}x \quad \text{for } v \in V \tag{4.10}$$

and relates to $|||\alpha|||$. The tangential component $\gamma_t(v)$ of some vector $v \in \mathbb{R}^n$ with respect to some normal vector ν reads

$$\gamma_t(v) := \begin{cases} v \cdot (0, -1; 1, 0)\nu & \text{if } n = 2, \\ v \times \nu & \text{if } n = 3 \end{cases}$$

and appears in the second (nonconsistency) residual

$$\operatorname{Res}_{\mathrm{NC}}(v) := \int_{\Gamma_D} v \cdot \gamma_t(\nabla u_D) \,\mathrm{d}s - \int_\Omega \nabla_{\mathrm{NC}} u_{\mathrm{CR}} \cdot \operatorname{Curl} v \,\mathrm{d}x \quad \text{for } v \in H^1_N(\Omega; \mathbb{R}^s), \tag{4.11}$$

which takes functions in

$$H^1_N(\Omega; \mathbb{R}^s) := \{ v \in H^1(\Omega; \mathbb{R}^s) \,\big|\, \operatorname{Curl} v \cdot \nu = 0 \text{ along } \Gamma_N \}.$$

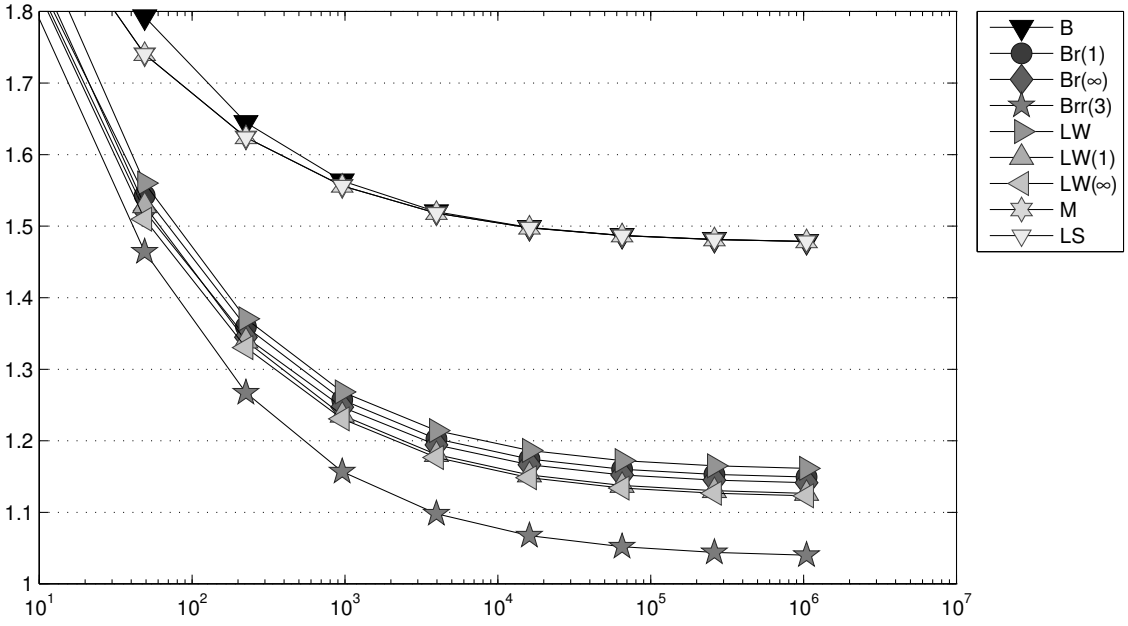

Figure 4.13: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.3.4.

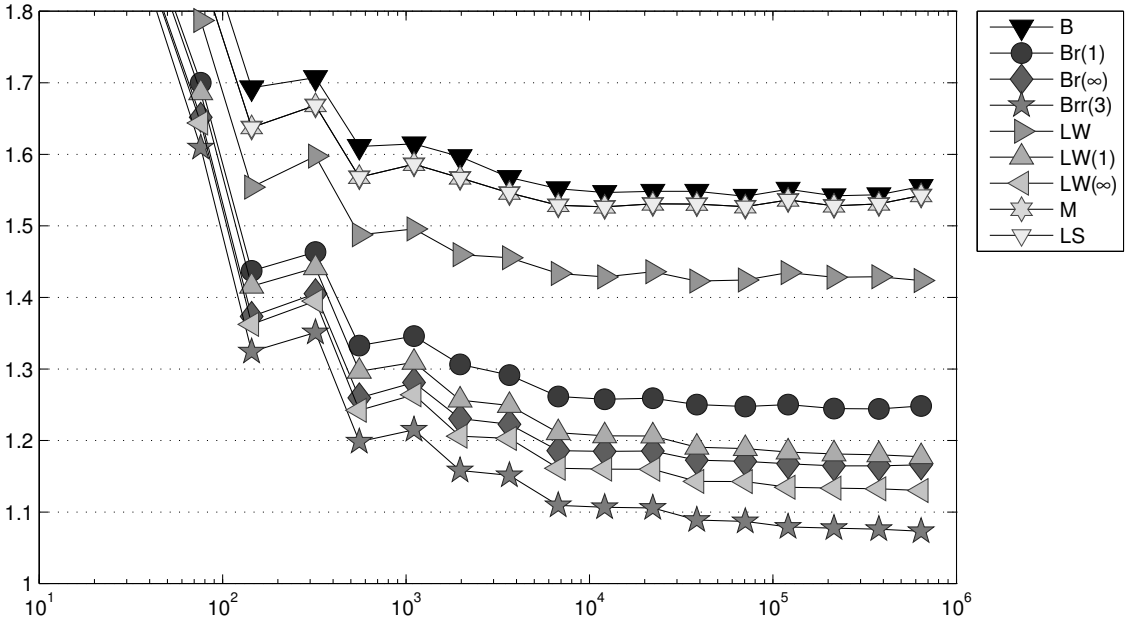

Figure 4.14: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.3.4.

This space relates to β in (4.9) and therefore the dual norm of $\mathrm{Res}_{\mathrm{NC}}$ is measured with respect to the weighted L^2 norm of the Curl, i.e.,

$$|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} := \sup_{\substack{v\in H^1_N(\Omega;\mathbb{R}^s)\\ \operatorname{Curl} v\neq 0}} \mathrm{Res}_{\mathrm{NC}}(v)/\left\|\mathbb{S}^{-1/2}\operatorname{Curl} v\right\|_{L^2(\Omega)}.$$

Theorem 4.4.1. *(a) For simply connected domains Ω, the energy norm of the error splits into*

$$|||e|||_{\mathrm{NC}}^2 = |||\alpha|||^2 + \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}^2 = |||\mathrm{Res}|||_\star^2 + |||\mathrm{Res}_{\mathrm{NC}}|||^2_{H^1_N(\Omega;\mathbb{R}^s)^\star}$$

with α and β from (4.9)*, and the residuals* Res *from* (4.10) *and* $\mathrm{Res}_{\mathrm{NC}}$ *from* (4.11)*.*
(b) There exists the alternative characterisation

$$|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} = \min_{\substack{v\in H^1(\Omega)\\ v=u_D \text{ on } \Gamma_D}} |||u_{\mathrm{CR}} - v|||_{\mathrm{NC}}.$$

(c) The quantity

$$\eta_0^2 := \sum_{T\in\mathcal{T}}\Big(\mathrm{C_P}(T)\left\|\lambda_{\min,T}^{-1/2}h_T(f-f_T)\right\|_{L^2(T)} + \left\|f_{\mathcal{T}}/n\ \mathbb{S}^{-1/2}(\bullet-\operatorname{mid}(T))\right\|_{L^2(T)} + \sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)}\mathrm{C_N}(E)\left\|\lambda_{\min,T}^{-1/2}h_E^{1/2}(g-g_E)\right\|_{L^2(E)}\Big)^2 \quad (4.12)$$

is a reliable and efficient error estimator for $|||\mathrm{Res}|||_\star$ in the sense that

$$|||\alpha|||^2 \leqslant \eta_0^2 \lesssim |||e|||_{\mathrm{NC}}^2 + \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2}f,\mathcal{T})^2 + \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2}g,\mathcal{E}(\Gamma_N))^2.$$

Proof. Proof of (a). The first equality directly follows from the orthogonality properties of the Helmholtz decomposition in Theorem 2.1.12. An integration by parts for any $v\in V$ and Theorem 2.1.12.(a) show

$$\mathrm{Res}(v) = \int_\Omega \nabla_{\mathrm{NC}}e\cdot\nabla v\,\mathrm{d}x = \int_\Omega \nabla\alpha\cdot\nabla v\,\mathrm{d}x \leqslant |||\alpha|||\,|||v|||.$$

This implies $|||\mathrm{Res}|||_\star \leqslant |||\alpha|||$. Moreover, $v=\alpha$ leads to $|||\mathrm{Res}|||_\star = |||\alpha|||$.

Another integration by parts for any $v\in H^1(\Omega;\mathbb{R}^s)$ and Theorem 2.1.12.(b) lead to

$$\begin{aligned}\mathrm{Res}_{\mathrm{NC}}(v) &= \int_{\Gamma_D} v\cdot\gamma_t(\nabla u_D)\,\mathrm{d}s - \int_\Omega \nabla_{\mathrm{NC}}u_{\mathrm{CR}}\cdot\operatorname{Curl} v\,\mathrm{d}x\\ &= \int_\Omega \nabla_{\mathrm{NC}}e\cdot\operatorname{Curl} v\,\mathrm{d}x\\ &= \int_\Omega (\mathbb{S}^{-1/2}\operatorname{Curl}\beta)\cdot(\mathbb{S}^{-1/2}\operatorname{Curl} v)\,\mathrm{d}x\\ &\leqslant \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}\left\|\mathbb{S}^{-1/2}\operatorname{Curl} v\right\|_{L^2(\Omega)}.\end{aligned}$$

This implies $\|\!|\mathrm{Res}_{\mathrm{NC}}|\!\|_{H^1_N(\Omega;\mathbb{R}^s)^\star} \leqslant \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}$ and $v = \beta$ leads to the identity $\|\!|\mathrm{Res}_{\mathrm{NC}}|\!\|_{H^1_N(\Omega;\mathbb{R}^s)^\star} = \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}$ which concludes the proof of (a).
Proof of (b). Given any $v \in H^1(\Omega)$ with $u_D - v = 0$ along Γ_D, Theorem 2.1.12.(a) yields

$$\begin{aligned}\left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|^2_{L^2(\Omega)} &= \int_\Omega \operatorname{Curl}\beta\cdot\nabla_{\mathrm{NC}} e\,\mathrm{d}x = \int_\Omega \operatorname{Curl}\beta\cdot(\nabla_{\mathrm{NC}}u_{\mathrm{CR}} - \nabla v)\,\mathrm{d}x\\ &\leqslant \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}\left\|\mathbb{S}^{1/2}(\nabla_{\mathrm{NC}}u_{\mathrm{CR}} - \nabla v)\right\|_{L^2(\Omega)}.\end{aligned}$$

Therefore,

$$\left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)} \leqslant \min_{\substack{v\in H^1(\Omega)\\ v=u_D \text{ on } \Gamma_D}} \left\|\mathbb{S}^{1/2}(\nabla_{\mathrm{NC}}u_{\mathrm{CR}} - \nabla v)\right\|_{L^2(\Omega)}.$$

By (4.9), the choice $v := u - \alpha$ leads to $\mathbb{S}^{1/2}(\nabla_{\mathrm{NC}}u_{\mathrm{CR}} - \nabla v) = -\mathbb{S}^{-1/2}\operatorname{Curl}\beta$ and minimises the right-hand side. This concludes the proof of (b).
Proof of (c). Consider the nonconforming interpolation $\alpha_{\mathrm{CR}} \in \mathrm{CR}(\mathcal{T})$ of α from Definition 2.2.15. An integration by parts yields

$$\int_T \nabla(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x = 0 \quad \text{for all } T \in \mathcal{T}. \tag{4.13}$$

This proves

$$\begin{aligned}\left\|\mathbb{S}^{1/2}\nabla(\alpha - \alpha_{\mathrm{CR}})\right\|^2_{L^2(T)} &= \int_T \mathbb{S}\nabla\alpha\cdot\nabla\alpha\,\mathrm{d}x - 2\int_T \mathbb{S}\nabla\alpha\cdot\nabla\alpha_{\mathrm{CR}}\,\mathrm{d}x + \int_T \mathbb{S}\nabla\alpha_{\mathrm{CR}}\cdot\nabla\alpha_{\mathrm{CR}}\,\mathrm{d}x\\ &= \int_T \mathbb{S}\nabla\alpha\cdot\nabla\alpha\,\mathrm{d}x - \int_T \mathbb{S}\nabla\alpha_{\mathrm{CR}}\cdot\nabla\alpha_{\mathrm{CR}}\,\mathrm{d}x\,\mathrm{d}x \leqslant \left\|\mathbb{S}^{1/2}\nabla\alpha\right\|^2_{L^2(T)}.\end{aligned}$$

With (4.8) and (4.13) the energy norm $\|\!|\alpha|\!\|^2$ equals

$$\begin{aligned}\|\!|\alpha|\!\|^2 = \int_\Omega (\mathbb{S}\nabla\alpha)\cdot\nabla_{\mathrm{NC}} e\,\mathrm{d}x &= \int_\Omega (\mathbb{S}\nabla\alpha)\cdot\nabla u\,\mathrm{d}x - \int_\Omega (\mathbb{S}\nabla\alpha_{\mathrm{CR}})\cdot\nabla_{\mathrm{NC}}u_{\mathrm{CR}}\,\mathrm{d}x\\ &= \int_\Omega f(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x + \int_{\Gamma_N} g(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}s\\ &= \int_\Omega (f - f_{\mathcal{T}})(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x + \int_{\Gamma_N}(g - g_{\mathcal{E}})(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}s\\ &\quad + \int_\Omega f_{\mathcal{T}}(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x + \int_{\Gamma_N} g_{\mathcal{E}}(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}s.\end{aligned}$$

The last term vanishes due to $\int_E \alpha - \alpha_{\mathrm{CR}}\,\mathrm{d}s = 0$ for all $E \in \mathcal{E}$ by definition of α_{CR}. The last argument and the trace identity (Theorem 2.2.17) for $v = \alpha - \alpha_{\mathrm{CR}}$ show

$$\begin{aligned}\int_T f_T(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x &= -\frac{1}{n}\int_\Omega f_T(x - \mathrm{mid}(T))\cdot\nabla_{\mathrm{NC}}(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x\\ &\leqslant \left\|f_{\mathcal{T}}/n\,\mathbb{S}^{-1/2}(\bullet - \mathrm{mid}(T))\right\|_{L^2(T)}\left\|\mathbb{S}^{1/2}\nabla(\alpha - \alpha_{\mathrm{CR}})\right\|_{L^2(T)}.\end{aligned}$$

A sum over all $T \in \mathcal{T}$ leads to

$$\int_\Omega f_\mathcal{T}(\alpha - \alpha_{\mathrm{CR}})\,\mathrm{d}x \leqslant \sum_{T\in\mathcal{T}} \left\| f_\mathcal{T}/n\, \mathbb{S}^{-1/2}(\bullet - \mathrm{mid}(T))\right\|_{L^2(T)} \left\|\mathbb{S}^{1/2}\nabla(\alpha - \alpha_{\mathrm{CR}})\right\|_{L^2(T)}. \quad (4.14)$$

The orthogonality of $f - f_\mathcal{T}$ onto $\mathcal{P}_0(\mathcal{T})$ allows the subtraction of the piecewise integral mean $v_\mathcal{T}$ of $v := \alpha - \alpha_{\mathrm{CR}}$. An elementwise Poincaré inequality yields

$$\begin{aligned}\int_\Omega (f - f_T)v\,\mathrm{d}x &= \sum_{T\in\mathcal{T}} \int_T (f - f_T)(v - v_T)\,\mathrm{d}x \\ &\leqslant \sum_{T\in\mathcal{T}} \mathrm{C_P}(T) h_T \left\|\lambda_{\min,T}^{-1/2}(f - f_T)\right\|_{L^2(T)} \left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}. \end{aligned} \quad (4.15)$$

Similar arguments for any $E \in \mathcal{E}(\Gamma_N) \cap \mathcal{E}(T)$ lead to

$$\int_E (g - g_E)v\,\mathrm{d}s \leqslant \mathrm{C_N}(E) \left\|\lambda_{\min,T}^{-1/2} h_E^{1/2}(g - g_E)\right\|_{L^2(E)} \left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}.$$

The sum over all $E \in \mathcal{E}(\Gamma_N)$ yields

$$\int_{\Gamma_N} (g - g_\mathcal{E})v\,\mathrm{d}s \leqslant \sum_{T\in\mathcal{T}} \left(\sum_{E\in\mathcal{E}(\Gamma_N)\cap\mathcal{E}(T)} \mathrm{C_N}(E) \left\|\lambda_{\min,T}^{-1/2} h_E^{1/2}(g - g_E)\right\|_{L^2(E)} \right) \left\|\mathbb{S}^{1/2}\nabla v\right\|_{L^2(T)}. \quad (4.16)$$

The sum of (4.14), (4.15), (4.16) and a Cauchy inequality in $\mathbb{R}^{|\mathcal{T}|}$ result in

$$|||\alpha||| = \int_\Omega (\mathbb{S}\nabla\alpha)\cdot\nabla_{\mathrm{NC}} e\,\mathrm{d}x \leqslant \eta_0 \left(\sum_{T\in\mathcal{T}} \left\|\mathbb{S}^{1/2}\nabla(\alpha - \alpha_{\mathrm{CR}})\right\|_{L^2(T)}^2 \right)^{1/2} \leqslant \eta_0\, |||\alpha|||.$$

This leads to the assertion $|||\alpha||| \leqslant \eta_0$. For a proof of efficiency, observe that

$$\left\| f_\mathcal{T}/n\, \mathbb{S}^{-1/2}(\bullet - \mathrm{mid}(\mathcal{T}))\right\|_{L^2(\Omega)} \leqslant \left\| f_\mathcal{T} h_\mathcal{T} \lambda_{\min,\mathcal{T}}^{-1/2}\right\|_{L^2(\Omega)}/n$$

and apply Theorem 3.4.5. □

In $n = 2$ dimensions, the Curl is just a rotation of the gradient ∇. Therefore, the residual (4.11) transforms to

$$\mathrm{Res_{NC}}(v) = \int_{\Gamma_D} v(\partial u_D/\partial s)\,\mathrm{d}s - \int_\Omega \mathrm{Curl_{NC}}\, u_{\mathrm{CR}} \cdot \nabla v\,\mathrm{d}x \quad \text{for } v \in H^1_N(\Omega).$$

This form of the residual applies to the prerequisits of Chapter 3 for interchanged boundaries Γ_D and Γ_N and data $f \equiv 0$, $g \equiv \partial u_D/\partial s$ and $\sigma_h := \mathrm{Curl_{NC}}\, u_{\mathrm{CR}}$. Since

$\|\mathrm{Curl}\,\cdot\|_{L^2(\Omega)} = \|\nabla\cdot\|_{L^2(\Omega)}$, the dual norm of $\mathrm{Res}_{\mathrm{NC}}$ equals

$$|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} = \sup_{\substack{v\in H^1_N(\Omega)\\ \nabla v\neq 0}} \mathrm{Res}(v)/\left\|\mathbb{S}^{-1/2}\nabla v\right\|_{L^2(\Omega)}.$$

The following lemma proves kernel properties of Res and $\mathrm{Res}_{\mathrm{NC}}$ that allows the application of any error estimator η or μ from Chapter 3, even those that solve local problems, to bound

$$|||\mathrm{Res}|||_\star \leqslant \eta \quad \text{and} \quad |||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} \leqslant \mu.$$

Lemma 4.4.2. *It holds*

$$\mathrm{Res}(\varphi_z) = 0 \quad \textit{for all free nodes } z\in\mathcal{M} \quad \textit{and} \quad \mathrm{Res}_{\mathrm{NC}}(\varphi_z) = 0 \quad \textit{for all nodes } z\in\mathcal{N}.$$

Proof. The first assertion follows directly from $V(\mathcal{T})\subset \mathrm{CR}_0(\mathcal{T})$ and (4.8). For the second assertion, an elementwise integration by parts shows, for any $z\in\mathcal{N}$,

$$\begin{aligned}
\mathrm{Res}_{\mathrm{NC}}(\varphi_z) &= \int_{\Gamma_D}\varphi_z\cdot\gamma_t(\nabla u_D)\,\mathrm{d}s - \int_\Omega \nabla u_{\mathrm{CR}}\cdot\mathrm{Curl}\,\varphi_z\,\mathrm{d}x\\
&= \sum_{T\in\mathcal{T}(z)}\int_{\Gamma_D\cap\partial T} u_D\,\mathrm{Curl}\,\varphi_z\cdot\nu\,\mathrm{d}s - \int_{\partial T} u_{\mathrm{CR}}\,\mathrm{Curl}\,\varphi_z\cdot\nu_T\,\mathrm{d}s\\
&= \sum_{T\in\mathcal{T}(z)}\sum_{E\in\mathcal{E}(T)}\left(\fint_{\Gamma_D\cap E} u_D\,\mathrm{d}s - u_{\mathrm{CR}}(\mathrm{mid}(E))\right)\int_E \mathrm{Curl}\,\varphi_z\cdot\nu_T\,\mathrm{d}s\\
&= \sum_{E\in\mathcal{E}(z)\cap\mathcal{E}(\Gamma_D)}\left(\fint_E u_D\,\mathrm{d}s - u_{\mathrm{CR}}(\mathrm{mid}(E))\right)\int_E \mathrm{Curl}\,\varphi_z\cdot\nu_T\,\mathrm{d}s\\
&\qquad - \sum_{E\in\mathcal{E}(z)\setminus\mathcal{E}(\Gamma_D)} u_{\mathrm{CR}}(\mathrm{mid}(E))\int_E [\mathrm{Curl}\,\varphi_z\cdot\nu_E]_E\,\mathrm{d}s.
\end{aligned}$$

The last equation is true, because u_{CR} is continuous in $\mathrm{mid}(\mathcal{E})$. Since

$$u_{\mathrm{CR}}(\mathrm{mid}(E)) = u_{D,\mathrm{CR}}(\mathrm{mid}(E)) = \fint_E u_D\,\mathrm{d}s \quad \text{for all } E\in\mathcal{E}(\Gamma_D),$$

the first sum vanishes. The second sum also vanishes, because $\mathrm{Curl}\,\varphi_z\in\mathrm{RT}_0(\mathcal{T}(z))$ has no normal jumps on interior edges $E\in\mathcal{E}(z)$. Hence, $\mathrm{Res}_{\mathrm{NC}}(\varphi_z) = 0$. □

However, in $n = 3$ dimensions the identity $\|\mathrm{Curl}\,\cdot\|_{L^2(\Omega)} = \|\nabla\cdot\|_{L^2(\Omega)}$ does not hold and $|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star}$ needs a different treatment. Subsection 4.4.2 shows an alternative approach that is applicable in any dimension.

Listing 4.1 computes $\left\|f_{\mathcal{T}}/2\,\mathbb{S}^{-1/2}\,(\bullet - \mathrm{mid}(T))\right\|_{L^2(\Omega)}$ in η_0 from Theorem 4.4.1 for $n = 2$ dimensions. The code employs the analytic formula of Lemma 2.2.19 in Line 11. The diffusion tensor $\mathbb{S}$ is assumed to be a constant scalar value on each element, which is stored in the vector `alpha4e`. The oscillation terms are computed separately with the

```
function [eta4e,eta] = estimateCREtaR(f,alpha4e,c4n,n4e,degree_f)
area4e = computeArea4e(c4n,n4e);
if nargin(f) == 1
    f = @(n4p,Gpts4p,Gpts4ref)(f(Gpts4p));
end
mean4e = integrate(c4n, n4e, f,degree_f);
mean4e = mean4e./(area4e*ones(1,size(mean4e,2)));
n4s = computeN4s(n4e);
length4s = computeLength4s(c4n,n4s);
s4e = computeS4e(n4e);
eta4e = sum(length4s(s4e).^2,2).*sum(mean4e.^2,2).*area4e./alpha4e/144;
eta = sqrt(sum(eta4e));
end
```

Listing 4.1: Listing of `estimateCRetaR.m`

functions of Appendix A.10.

4.4.2 Alternative Estimation of the Nonconsistency Residual

Theorem 4.4.1.(b) provides another way for the estimation of $\|\!|\mathrm{Res}_{\mathrm{NC}}|\!\|_{H^1_N(\Omega;\mathbb{R}^s)^\star}$ by the design of conforming functions $v_{\mathrm{xyz}} \in H^1(\Omega)$ with $v_{\mathrm{xyz}} = u_D$ along Γ_D, such that

$$\|\!|\mathrm{Res}_{\mathrm{NC}}|\!\|_{H^1_N(\Omega;\mathbb{R}^s)^\star} \leqslant \|\!|u_{\mathrm{CR}} - v_{\mathrm{xyz}}|\!\|_{\mathrm{NC}} =: \mu_{\mathrm{xyz}}. \tag{4.17}$$

The following subsections discuss efficient schemes to compute such suitable v_{xyz}.

4.4.2.1 Interpolation by Ainsworth

Ainsworth (2004) designs some piecewise linear $v_{\mathrm{A}} \in u_{D,h} + V(\mathcal{T})$ by

$$v_{\mathrm{A}}(z) \coloneqq \begin{cases} u_D(z) & \text{if } z \in \mathcal{N}(\Gamma_D), \\ \left(\sum_{T\in\mathcal{T}(z)} \lambda_{\max,T}^{1/2} u_{\mathrm{CR}}|_T(z)\right) / \left(\sum_{T\in\mathcal{T}(z)} \lambda_{\max,T}^{1/2}\right) & \text{if } z \in \mathcal{M}. \end{cases} \tag{4.18}$$

The error estimator reads

$$\mu_{\mathrm{A}} \coloneqq \|\!|u_{\mathrm{CR}} - v_{\mathrm{A}}|\!\|_{\mathrm{NC}}.$$

For a proof of its efficiency in the sense of $\mu_{\mathrm{A}} \lesssim \left\|\mathbb{S}^{-1/2}\operatorname{Curl}\beta\right\|_{L^2(\Omega)}$, see Ainsworth (2004, Theorem 6.4).

The improved interpolation of (Vohralík, 2007; Ainsworth, 2007/08; Braess, 2009) employs the auxiliary function

$$v^0 \coloneqq u_{\mathrm{CR}} - f_T\psi/n \quad \text{with}$$
$$\psi(x) \coloneqq \frac{1}{2}(x-\mathrm{mid}(T))^T\mathbb{S}^{-1}(x-\mathrm{mid}(T)) - \fint_T (y-\mathrm{mid}(T))^T\mathbb{S}^{-1}(y-\mathrm{mid}(T))\,\mathrm{d}y$$
$$\text{for } x \in T \in \mathcal{T}.$$

An averaging of $v^0 \in \mathcal{P}_2(\mathcal{T})$ as above leads to some piecewise quadratic and continuous

function $v_{\mathrm{AP2}} \in \mathcal{P}_2(\mathcal{T}) \cap C(\Omega)$. To define a piecewise quadratic continuous functions in 2D, one has to assign the nodal values for the boundary nodes $\mathcal{N}_2(\Gamma_D) \coloneqq \mathcal{N}(\Gamma_D) \cup \mathrm{mid}(\mathcal{E}(\Gamma_D))$ and the remaining free nodes $\mathcal{M}_2 \coloneqq (\mathcal{N} \cup \mathrm{mid}(\mathcal{E}))\backslash\mathcal{N}_2(\Gamma_D)$. In the present design the nodal values read

$$v_{\mathrm{AP2}}(z) \coloneqq \begin{cases} u_D(z) & \text{if } z \in \mathcal{N}_2(\Gamma_D), \\ \left(\sum_{T\in\mathcal{T}(z)} \lambda_{\max,T}^{1/2} v^0|_T(z)\right) / \left(\sum_{T\in\mathcal{T}(z)} \lambda_{\max,T}^{1/2}\right) & \text{if } z \in \mathcal{M}_2. \end{cases}$$

Note that $\mathcal{T}(z)$ for $z \in \mathrm{mid}(\mathcal{E})$ contains only two elements. The error estimator reads

$$\mu_{\mathrm{AP2}} \coloneqq |||u_{\mathrm{CR}} - v_{\mathrm{AP2}}|||_{\mathrm{NC}}.$$

4.4.2.2 Modified Interpolation

The novel design from Carstensen and Merdon (2013) for $n = 2$ dimensions employs the red-refined triangulation $\mathrm{red}(\mathcal{T})$. At the boundary the interpolation equals the nodal interpolation of u_D and on all edge midpoints it equals u_{CR}, i.e.,

$$v_{\mathrm{RED}}(z) \coloneqq \begin{cases} u_{\mathrm{CR}}(z) & \text{for } z \in \mathrm{mid}(\mathcal{E})\backslash\,\mathrm{mid}(\mathcal{E}(\Gamma_D)), \\ u_D(z) & \text{for } z \in (\mathcal{N} \cup \mathrm{mid}(\mathcal{E})) \cap \Gamma_D, \\ v_z & \text{for } z \in \mathcal{M}. \end{cases} \tag{4.19}$$

The remaining values v_z for $z \in \mathcal{M}$ are chosen either by interpolation schemes as in the previous subsubsection or locally in an optimal way as follows. Consider the node patch $\omega_z^{\mathrm{red}} \coloneqq \{x \in \Omega \,|\, \varphi_z^{\mathrm{red}}(x) > 0\}$ of the nodal basis function φ_z^{red} of some $z \in \mathcal{M}$ with respect to the red-refined triangulation as in Figure 4.15. Since all boundary nodes along $\partial\omega_z^{\mathrm{red}}$ are fixed by (4.19), the only degree of freedom remains v_z. So we start with $v_z = 0$ and then compute the optimal value

$$\alpha_{\mathrm{PMRED}} = \operatorname*{argmin}_{\alpha\in\mathbb{R}} \left\| \mathbb{S}^{1/2} \left(\nabla_{\mathrm{NC}} u_{\mathrm{CR}} - \nabla(v_{\mathrm{RED}} + \alpha\varphi_z^{\mathrm{red}}) \right) \right\|_{L^2(\omega_z^{\mathrm{red}})}.$$

The choice $v_z = \alpha_{\mathrm{PMRED}}$ defines the piecewise minimal interpolation

$$v_{\mathrm{PMRED}}|_{\omega_z^{\mathrm{red}}} \coloneqq v_{\mathrm{RED}} + \alpha\varphi_z^{\mathrm{red}}.$$

The choice by averaging as in (4.18) is labelled v_{ARED}. Whatever the choice is, any v_{RED} equals u_{CR} on the central subtriangle of every red-refined triangle $T \in \mathcal{T}$ as depicted in Figure 4.15. The associated error estimators read

$$\mu_{\mathrm{ARED}} \coloneqq |||u_{\mathrm{CR}} - v_{\mathrm{ARED}}|||_{\mathrm{NC}} \quad \text{and} \quad \mu_{\mathrm{PMRED}} \coloneqq |||u_{\mathrm{CR}} - v_{\mathrm{PMRED}}|||_{\mathrm{NC}}.$$

Remark 4.4.3. *In two dimensions, the polynomial space $\mathcal{P}_2(\mathcal{T})$ is isomorph to $\mathcal{P}_1(\mathrm{red}(\mathcal{T}))$ in the sense that the set of linear functionals L in Definition 2.2.1 consists of the same point evaluations. Hence, any design of some $\mathcal{P}_2(\mathcal{T})$ function is also applicable for the design of some piecewise linear function in $\mathcal{P}_1(\mathrm{red}(\mathcal{T})) \cap C(\Omega)$ and vice versa. In 3D, the degrees of freedom lay on the edges and do not coincide with the degrees of freedom of Crouzeix-Raviart functions that are in the face*

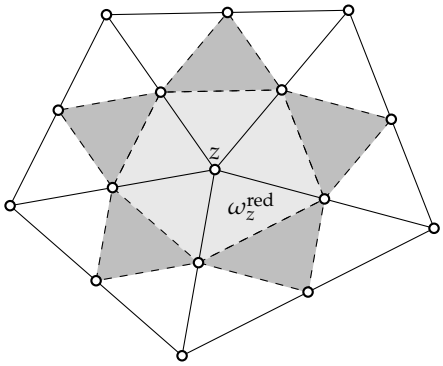

Figure 4.15: Node patch ω_z^{red} of some $z \in \mathcal{N}(\Omega)$ (light gray) with respect to the red-refined triangulation $\text{red}(\mathcal{T}(z))$ and central subtriangles (dark gray) that do not belong to any node patch of these node patches.

midpoints. In this case, some further averaging is necessary.

4.4.2.3 Optimal Interpolations

Since the nodal basis functions are included in $CR^1(\mathcal{T})$, the optimal $v_{\text{MP1}} \in u_{D,h} + V(\mathcal{T})$ equals the solution u_h of the conforming formulation of the Poisson problem from (4.4), i.e.,

$$F(v_h) = \int_\Omega \sigma_{\text{CR}} \cdot \nabla v_h \,\mathrm{d}x = \int_\Omega \sigma_h \cdot \nabla v_h \,\mathrm{d}x \quad \text{for all } v_h \in V(\mathcal{T}).$$

For comparison, we also compute the optimal $v_{\text{MP1RED}} \in \mathcal{P}_1(\text{red}(\mathcal{T})) \cap C(\Omega)$ on the red-refined triangulation $\text{red}(\mathcal{T})$ and the optimal piecewise quadratic $v_{\text{MP2}} \in \mathcal{P}_2(\mathcal{T}) \cap C(\Omega)$. Note that they do not have to equal the corresponding $\mathcal{P}_1$ conforming solutions of the Poisson problem. To reduce the computational costs of v_{MP1RED}, one might use $v_{\text{MP1REDcg(0)}} \coloneqq v_{\text{ARED}}$ as an initial guess for some iterative solver to approach the optimal value. The third iterate of a Jacobi preconditioned conjugate gradients algorithm defines $v_{\text{MP1REDcg(3)}}$. Similarly, $v_{\text{MP2cg(0)}} \coloneqq v_{\text{MP1REDcg(0)}}$ serves as an initial value to approach v_{MP2} and the third iterate of a Jacobi preconditioned conjugate gradients algorithm defines $v_{\text{MP2cg(3)}}$. The associated error estimators μ_{MP1}, μ_{MP2}, $\mu_{\text{MP2cg(3)}}$ μ_{MP1RED} and $\mu_{\text{MP1REDcg(3)}}$ are defined in the same way as μ_{A} in Subsubsection 4.4.2.1.

4.4.3 Modifications for Inhomogeneous Dirichlet Boundary Conditions

The designs of the test function v in the previous subsections may not resolve the inhomogeneous boundary data exactly. To qualify them as a valid upper bound in the sense of (4.17), we apply Theorem 4.2.2 for $v_D = u_D - v|_{\Gamma_D}$ and design some $w_D \in H^1(\Omega)$ with $w_D = v_D$ on Γ_D. Since w_D satisfies $u - (v + w_D) \in H^1_D(\Omega)$, (4.17) and the energy norm

estimate for w_D from Theorem 4.2.2 yield

$$\begin{aligned}|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} &\leqslant |||u_{\mathrm{CR}} - (v + w_D)|||_{\mathrm{NC}} \leqslant |||u_{\mathrm{CR}} - v|||_{\mathrm{NC}} + |||w_D||| \\ &\leqslant |||u_{\mathrm{CR}} - v|||_{\mathrm{NC}} + \mathrm{C}_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) \left\| h_{\mathcal{E}}^{3/2} \lambda_{\max,\mathcal{T}}^{1/2} \partial_{\mathcal{E}}^2 (u_D - v)/\partial s^2 \right\|_{L^2(\Gamma_D)}.\end{aligned}$$

For the more elaborate designs of v_{AP2}, v_{ARED}, v_{PMRED}, v_{MP1RED}, $v_{\mathrm{MP1REDcg(3)}}$,v_{MP2} or $v_{\mathrm{MP2cg(3)}}$ based on $\mathcal{P}_1(\mathrm{red}(\mathcal{T}))$ or $\mathcal{P}_2(\mathcal{T})$, the computation of w_D in Theorem 4.2.2 can be performed on $\mathrm{red}(\mathcal{T})$ with halved edge lengths, because $v_D \in H^1_0(E)$ for all Dirichlet boundary edges $E \in \mathcal{E}^{\mathrm{red}}(\Gamma_D)$ of $\mathrm{red}(\mathcal{T})$. This leads to the improved constant $\mathrm{C}_{\mathrm{D},1}(\mathcal{E}^{\mathrm{red}}(\Gamma_D)) = 0.4980/2^{3/2} = 0.1761$ for triangulations that consist of right isosceles triangles. This acknowledges the fact that the boundary data of these designs are already closer to u_D than the nodal interpolation on $\mathcal{T}$ as in the design of v_{A} in Subsubsection 4.4.2.1.

4.4.4 Connection Between Conforming Interpolation and Equilibration in 2D

The previous techniques of Subsubsections 4.4.2.1-4.4.2.3 design piecewise polynomial interpolations $v \in \mathcal{P}_1(\widehat{\mathcal{T}}) \cap C(\Omega)$ of u_{CR}. In the 2D case, the rotation of ∇v results in a divergence-free quantity $q \coloneqq \mathrm{Curl}\, v \in \mathrm{RT}_0(\widehat{\mathcal{T}})$ on some triangulation $\widehat{\mathcal{T}}$ (e.g. $\widehat{\mathcal{T}} = \mathrm{red}(\mathcal{T})$ for $v = v_{\mathrm{ARED}}$) with

$$|||u_{\mathrm{CR}} - v|||_{\mathrm{NC}} = \left\| \mathbb{S}^{1/2} \mathrm{Curl}_{\mathrm{NC}}(u_{\mathrm{CR}} - v) \right\|_{L^2(\Omega)}.$$

This leads to the upper bound

$$|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega;\mathbb{R}^s)^\star} \leqslant |||\gamma_{\Gamma_D}(\partial(u_D - v)/\partial s)|||_\star + \left\| \mathbb{S}^{1/2} \mathrm{Curl}_{\mathrm{NC}}(u_{\mathrm{CR}} - v) \right\|_{L^2(\Omega)}$$

and may offer different results in case of inhomogeneous boundary conditions. For example, if $v|_{\Gamma_D}$ satisfies the integral mean property $\int_E \partial(u_D - v)/\partial s \, \mathrm{d}s$ for all $E \in \mathcal{E}(\Gamma_D)$ (as v_{A} from Subsubsection 4.4.2.1), another Poincaré inequality yields

$$\begin{aligned}|||\gamma_{\Gamma_D}(\partial(u_D - v)/\partial s)|||_\star &\leqslant \mathrm{C}_{\mathrm{N}}(\mathcal{E}(\Gamma_D)) \left\| h_{\mathcal{E}}^{1/2} \lambda_{\min,\mathcal{T}}^{-1/2} \partial(u_D - v)/\partial s \right\|_{L^2(\Gamma_D)} \\ &\leqslant \frac{\mathrm{C}_{\mathrm{N}}(\mathcal{E}(\Gamma_D))}{\pi} \left\| h_{\mathcal{E}}^{3/2} \lambda_{\min,\mathcal{T}}^{-1/2} \partial_{\mathcal{E}}^2 (u_D - v)/\partial s^2 \right\|_{L^2(\Gamma_D)}\end{aligned}$$

with the trace constant $\mathrm{C}_{\mathrm{N}}(\mathcal{E}(\Gamma_D))$ similar to $\mathrm{C}_{\mathrm{N}}(\mathcal{E}(\Gamma_N))$ from Theorem 3.2.2. Since, on triangulations into right isosceles triangles, $\mathrm{C}_{\mathrm{N}}(\mathcal{E}(\Gamma_D))/\pi \leqslant 1.1474/\pi = 0.3652$ is smaller than $\mathrm{C}_{\mathrm{D},1}(\mathcal{E}(\Gamma_D)) = 0.4980$ from Theorem 4.2.2, the smaller value is used for the numerical experiments below. For designs that satisfy the integral mean property on the boundary of the red-refined triangulations, the constant reduces to $\mathrm{C}_{\mathrm{N}}(\mathcal{E}^{\mathrm{red}}(\Gamma_D)) = 0.3652/2^{3/2} = 0.1291$. Notice that $\partial^2 v/\partial s^2$ is zero for all piecewise affine designs, but not for the piecewise quadratic function v_{AP2} from Subsubsection 4.4.2.1. In the latter case, $\partial^2 v_{\mathrm{AP2}}/\partial s^2$ equals the second derivative of the $\mathcal{P}_2$ nodal interpolation.

ndof	8	40	176	736	3008	12160	...	785408
$\|\|\|e\|\|\|_{\mathrm{NC}}$	5.23e-02	5.19e-02	2.87e-02	1.98e-02	1.03e-02	5.17e-03	...	6.49e-04
$\mathrm{osc}(f)$	2.03e-01	9.44e-02	3.91e-02	9.38e-03	2.43e-03	6.13e-04	...	9.61e-06
η_0	2.11e-01	1.11e-01	5.21e-02	1.90e-02	7.69e-03	3.36e-03	...	3.68e-04
B	2.35e-01	1.38e-01	6.70e-02	2.13e-02	8.03e-03	3.40e-03	...	3.68e-04
M	2.32e-01	1.34e-01	6.61e-02	2.11e-02	7.99e-03	3.40e-03	...	3.68e-04
LS	2.32e-01	1.34e-01	6.61e-02	2.11e-02	7.99e-03	3.40e-03	...	3.68e-04
LW	2.22e-01	1.14e-01	4.85e-02	1.73e-02	7.12e-03	3.20e-03	...	3.65e-04
Br(1)	2.30e-01	1.30e-01	6.55e-02	2.05e-02	7.65e-03	3.22e-03	...	3.45e-04
Br(o)	2.27e-01	1.28e-01	6.50e-02	2.04e-02	7.63e-03	3.21e-03	...	3.45e-04
Bmr(1)	1.28e-01	7.27e-02	3.33e-02	1.36e-02	6.00e-03	2.83e-03	...	3.39e-04
LW(1)	2.20e-01	1.11e-01	4.72e-02	1.67e-02	6.80e-03	3.04e-03	...	3.45e-04
LW(o)	2.19e-01	1.10e-01	4.68e-02	1.66e-02	6.79e-03	3.04e-03	...	3.45e-04
LWm(1)	1.19e-01	6.14e-02	3.22e-02	1.30e-02	5.94e-03	2.84e-03	...	3.42e-04
Brr(3)	2.26e-01	1.25e-01	6.42e-02	2.01e-02	7.52e-03	3.16e-03	...	3.39e-04
Bmrr(3)	7.38e-02	4.48e-02	2.37e-02	1.13e-02	5.44e-03	2.68e-03	...	3.32e-04

Table 4.1: Guaranteed upper bounds for $\|\|\|\mathrm{Res}\|\|\|_\star$ by η_0 and the equilibration error estimators η_{M}, η_{LS}, η_{B}, η_{LW} and some of their postprocessings for uniform mesh refinement in the square example of Subsection 4.5.1 with respect to the number of degrees of freedom (ndof).

4.5 Numerical Experiments for Nonconforming CR-FEM

This section presents some numerical examples in order to compare the efficiency of all the error estimators η and μ of Section 4.4 in

$$|||e|||_{\mathrm{NC}}^2 \leqslant \eta^2 + \mu^2$$

for the estimation of $|||e|||_{\mathrm{NC}}$ via Theorem 4.4.1. First, Subsection 4.5.1 studies the efficiency of η_0 from Theorem 4.4.1.(c) compared to any other error estimator η. The remaining sections concern the efficiency of the complete guaranteed upper bound in some benchmark examples on domains depicted in Figure 4.2. The adaptive mesh refinement in all examples is driven by the Dörfler marking of Subsubsection 2.3.4.2 with the edge-based refinement indicators

$$\eta(E)^2 \coloneqq \begin{cases} h_E^2 \lambda_{\min,\omega_E}^{-1} \|f\|_{L^2(\omega_E)}^2 + h_E \lambda_{\max,E}^{-1} \|[\sigma_{\mathrm{CR}} \cdot \tau_E]\|_{L^2(E)}^2 & \text{for } E \in \mathcal{E}(\Omega), \\ h_E^2 \lambda_{\min,\omega_E}^{-1} \|f\|_{L^2(\omega_E)}^2 + 0.1334 \left\| h_E^{3/2} \lambda_{\min,E}^{-1/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(E)}^2 & \text{for } E \in \mathcal{E}(\Gamma_D), \\ h_E^2 \lambda_{\min,\omega_E}^{-1} \|f\|_{L^2(\omega_E)}^2 + h_E \lambda_{\max,E}^{-1} \|g - \sigma_{\mathrm{CR}} \cdot \nu_E\|_{L^2(E)}^2 & \text{for } E \in \mathcal{E}(\Gamma_N). \end{cases} \tag{4.20}$$

These refinement indicators are similar to the ones in Mao et al. (2010) and lead to a convergent adaptive method with optimal complexity. Other refinement indicators derived directly from the guaranteed upper bounds were tested in Carstensen and Merdon (2013) and led to very similar adaptively refined meshes and convergence histories. In all experiments, the bulk parameter for adaptive mesh refinement is $\theta = 0.5$.

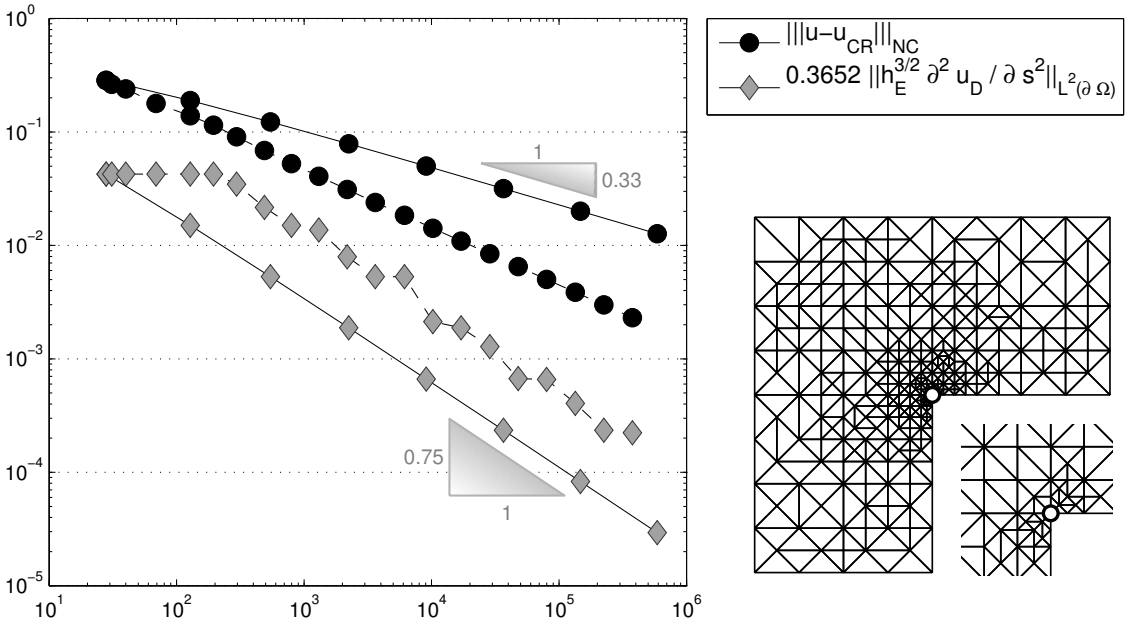

Figure 4.16: Convergence history for the energy error $|||e|||_{\mathrm{NC}}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.5.2 with respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$. The right image shows the adaptive mesh on level 8 and the neighbourhood of the singular point $(0,0)$ magnified by a factor 4.

4.5.1 Efficient Estimation of the Conforming Residual

The first experiment focuses on the efficient estimation of the dual norm $|||\mathrm{Res}|||_\star$ of the first residual Res in Theorem 4.4.1.(a) in the example of Subsubsection 4.3.2. Theorem 4.4.1.(c) provides a cheap and explicit upper bound η_0. Alternatively, one can employ any equilibration error estimator from Chapter 3.

Table 4.1 lists η_0 and other guaranteed upper bounds of $|||\mathrm{Res}|||_\star$ for several equilibration error estimators.

The overall conclusion is that there is little to no improvement by more elaborate error estimators like η_{B} or η_{LW}. Moreover, there is no improvement of η_0 or η_{B} by η_{M} on finer meshes. At least on coarse meshes with large oscillations of f, the postprocessed and mean-corrected error estimators $\eta_{\mathrm{Bmr(1)}}$, $\eta_{\mathrm{Bmrr(3)}}$ and $\eta_{\mathrm{LWm(1)}}$ lead to more significant reductions compared to η_0. On the finest mesh, $\eta_{\mathrm{Bmrr(3)}}$ is about 10 percent smaller than η_0.

4.5.2 L-Shaped domain

The first benchmark problem employs $f \equiv 0$, $\mathbb{S} \equiv \mathbb{I}$ and inhomogeneous Dirichlet data u_D on $\Gamma_D = \partial\Omega$ given by the exact solution

$$u(r,\varphi) = r^{2/3}\sin(2\varphi/3)$$

in polar coordinates on the L-shaped domain $\Omega = (-1,1)^2 \setminus ([0,1]\times[-1,0])$. The problem involves a typical corner singularity and shows an empirical convergence rate of $1/3$ with

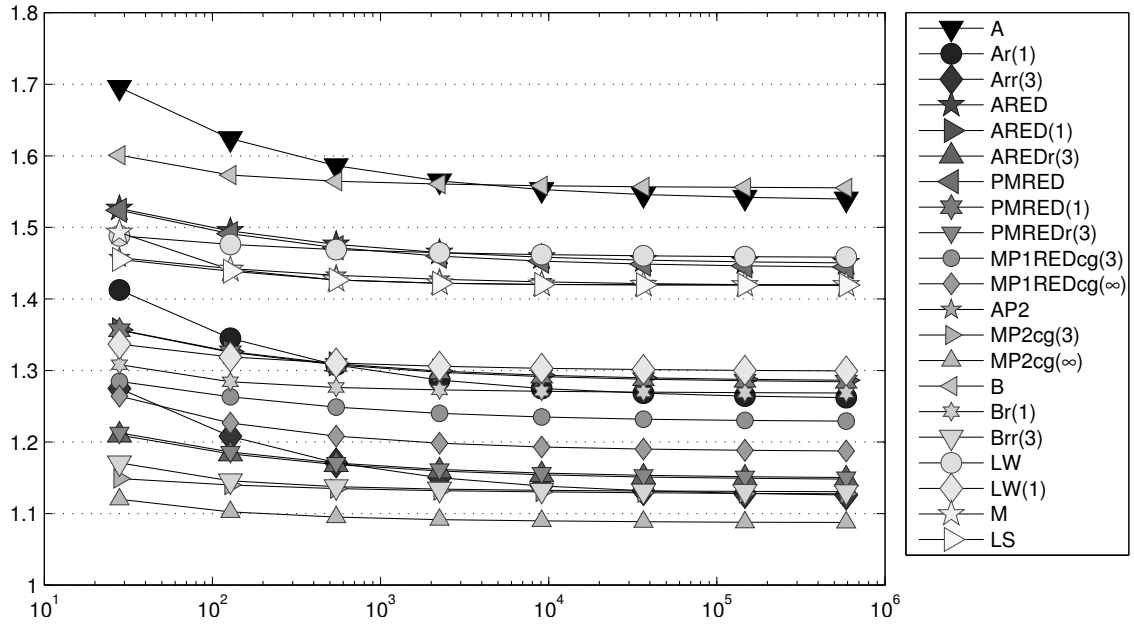

Figure 4.17: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.5.2.

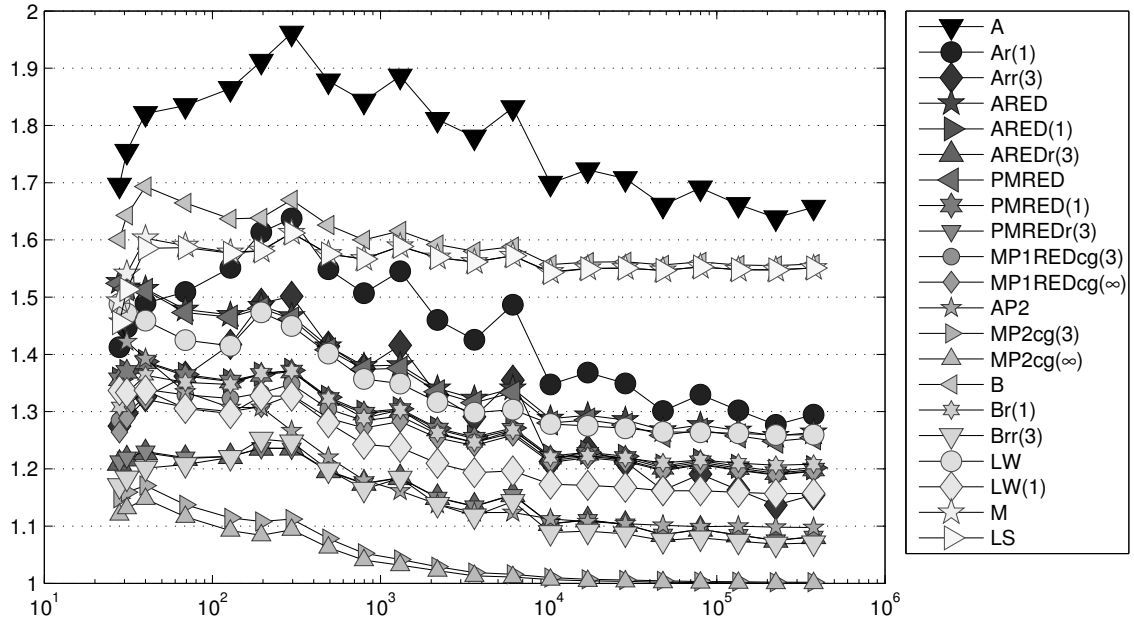

Figure 4.18: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.5.2.

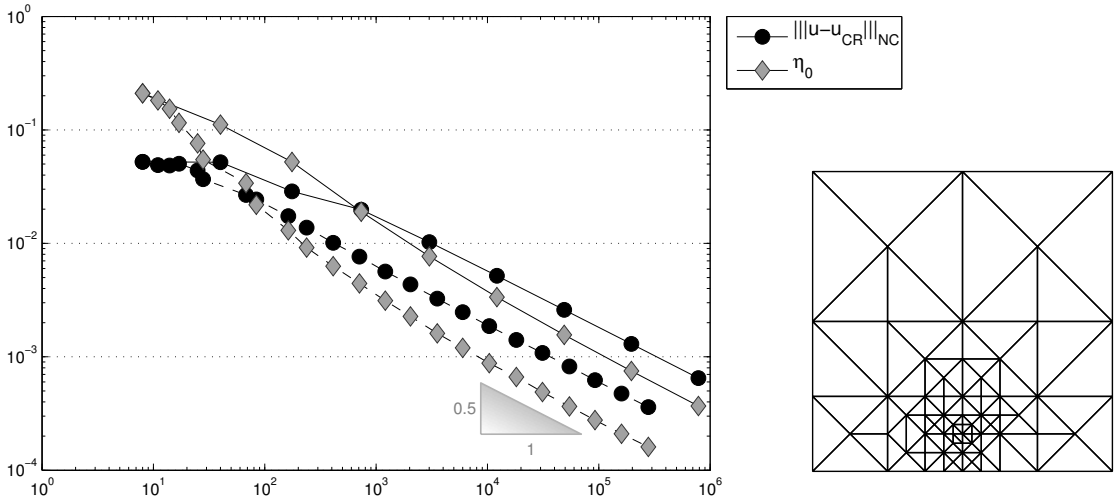

Figure 4.19: Convergence history for the energy error $|||e|||_{\mathrm{NC}}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.5.3 with respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$. The right image shows the adaptive mesh on level 8.

respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$ for uniform mesh refinement. Since the source term is zero, the overhead contribution η_0 of the error estimators vanish. The adaptive algorithm refines towards the singularity as depicted in Figure 4.16 and thus leads to the optimal convergence rate of $1/2$.

Figures 4.17 and 4.18 display the efficiency indices of all error estimators. They range from 1 for η_{MP2} to 1.7 for η_{A}. The novel interpolations on the red-refinement η_{ARED} and η_{PMRED} perform well with efficiency indices below 1.3 for adaptive mesh refinement. However, the quadratic interpolation η_{AP2} is slightly more efficient. While the performance of η_{B} is comparable to η_{A} for uniform mesh refinement, it is slightly better than η_{A} for adaptive mesh refinement. The Luce-Wohlmuth error estimator η_{LW} without postprocessing assumes indices around 1.5 and $\eta_{\mathrm{LW}(1)}$ with active postprocessing improves it to 1.3. The postprocessed Braess equilibration error estimator $\eta_{\mathrm{Brr}(3)}$ even leads to efficiency indices around 1.1. The other postprocessed error estimators show similar improvements.

4.5.3 Square with Large Oscillations

This subsection revisits the square domain example of Subsections 4.3.2 and 4.5.1. Figure 4.19 shows the convergence history for the broken energy error $|||e|||_{\mathrm{NC}}$ and η_0 as well as an adaptively refined mesh that looks similar to that computed by the conforming $\mathcal{P}_1$-FEM with the same conclusions with respect to the shortened pre-asymptotic range.

Although the error contribution η_0 is about 60 percent of $|||e|||_{\mathrm{NC}}$, the efficiency indices in Figures 4.20 and 4.21 are as good as in Subsection 4.5.2 with $\eta_0 = 0$. This leads to the conclusion that η_0 is a very sharp approximation of $|||\mathrm{Res}|||_\star$ and as such is not a critical term. Moreover, it seems to make sense to put more effort in the estimation of $|||\mathrm{Res}_{\mathrm{NC}}|||_{H^1_N(\Omega)^\star}$. For this, the postprocessed equilibration error estimators $\eta_{\mathrm{Bmrr}(3)}$ or the optimal quadratic interpolation $\eta_{\mathrm{MP2cg}(3)}$ are the best choice. In this example, only for uniform mesh refinement, the error estimator η_{AP2} is slightly worse than η_{ARED} and

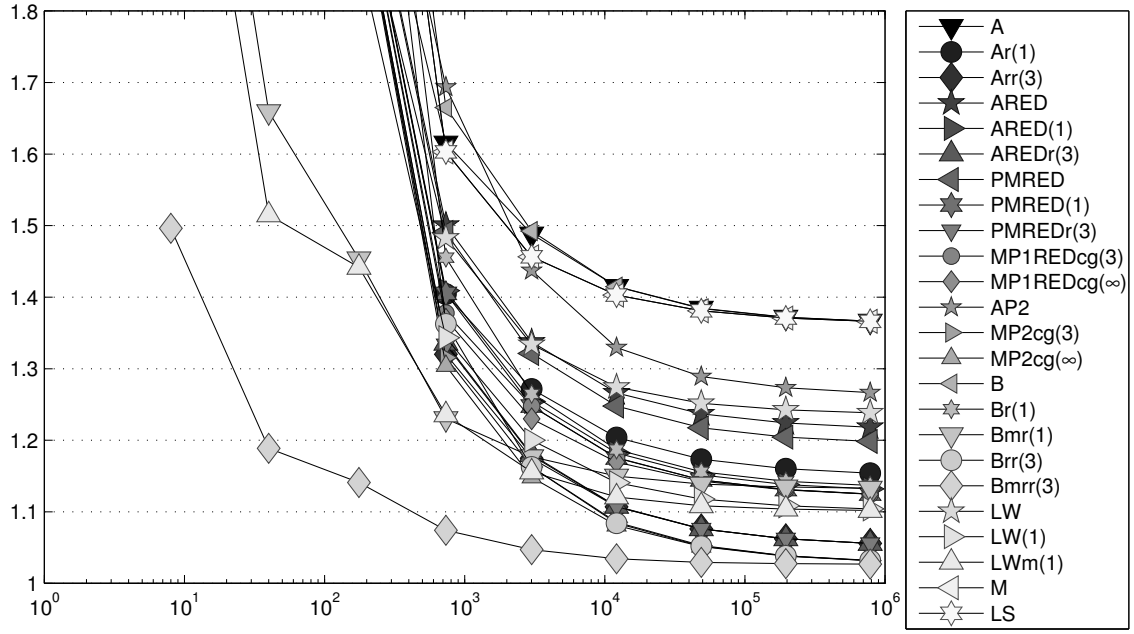

Figure 4.20: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.5.3.

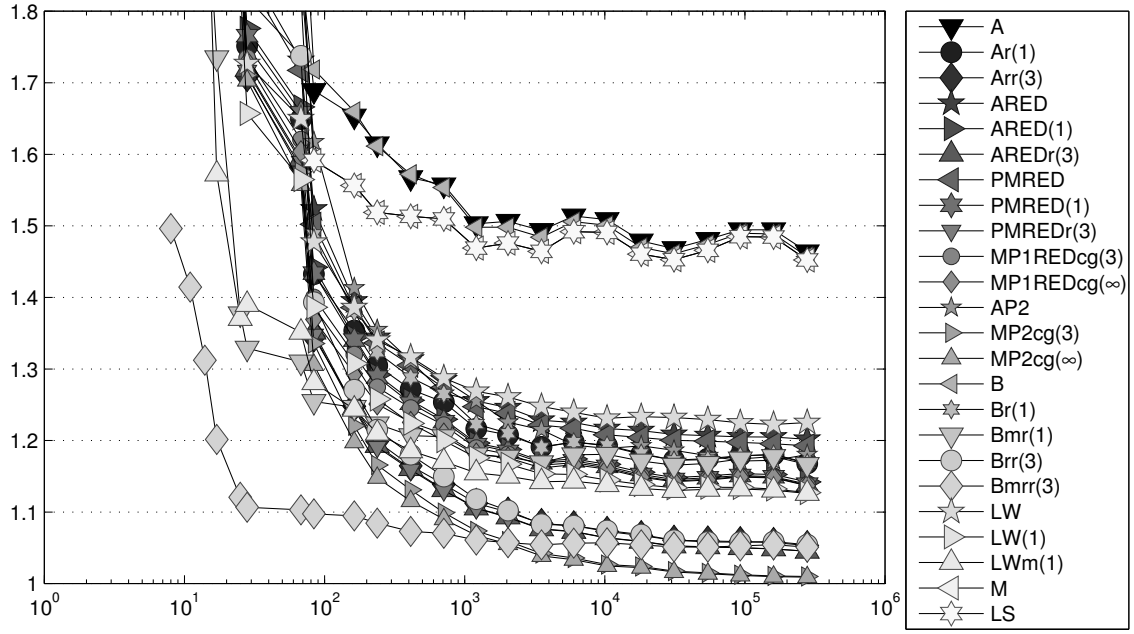

Figure 4.21: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.5.3.

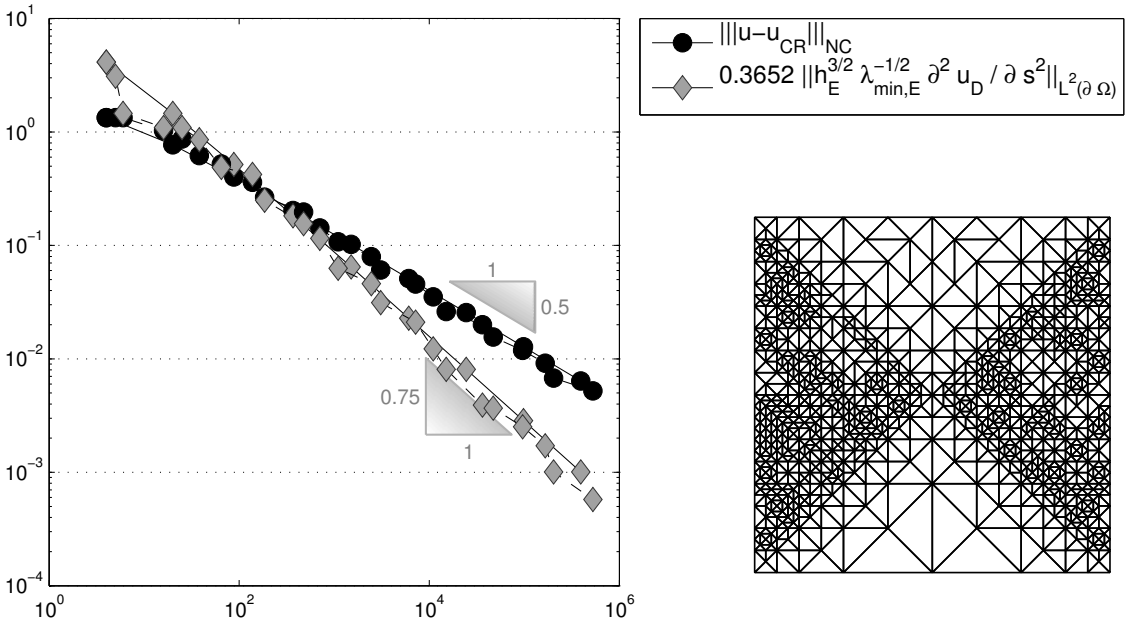

Figure 4.22: Convergence history for the energy error $|||e|||_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \lambda_{\min,\mathcal{E}}^{-1/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.5.4 with respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$. The right image shows the adaptive mesh on level 12.

η_{PMRED}.

4.5.4 Square with Discontinuous Diffusion Coefficients

This subsection revisits the square domain example of Subsection 4.3.3. Since $f \equiv 0$, the contribution η_0 vanishes. Figure 4.22 shows the convergence history for the broken energy norm and the Dirichlet error contribution that is of higher order but decreases slower than the oscillations in the last example. This especially affects the efficiency indices of η_{A} and η_{B} due to their coarse Dirichlet boundary approximation (see Figures 4.23 and 4.24). Estimators based on a red-refinement like η_{ARED} or η_{PMRED} allow for a lower constant in front of the Dirichlet error contribution and are therefore less affected by its influence. The postprocessed estimators $\eta_{\mathrm{Brr}(3)}$ or $\eta_{\mathrm{AREDr}(3)}$ assume efficiency indices around 1.1 as in the other examples.

4.5.5 Octagon with Discontinuous Diffusion Coefficients

This subsection revisits the square domain example of Subsection 4.3.4. Since $f \equiv 0$, the contribution η_0 vanishes. The efficiency indices from Figure 4.26 and 4.27 are similar to the results from Subsection 4.5.4. The error estimator $\eta_{\mathrm{MP2}(\infty)}$ is asymptotically exact, because the solution u is a quadratic polynomial and only the extra term that measures the inhomogeneous Dirichlet boundary condition is inexact. Moreover, this example shows that the error estimator $\eta_{\mathrm{MP2}(3)}$ with three pcg steps to approximate $\eta_{\mathrm{MP2}(\infty)}$ assumes efficiency indices around 1.15 and is not as close as in the examples without jumping

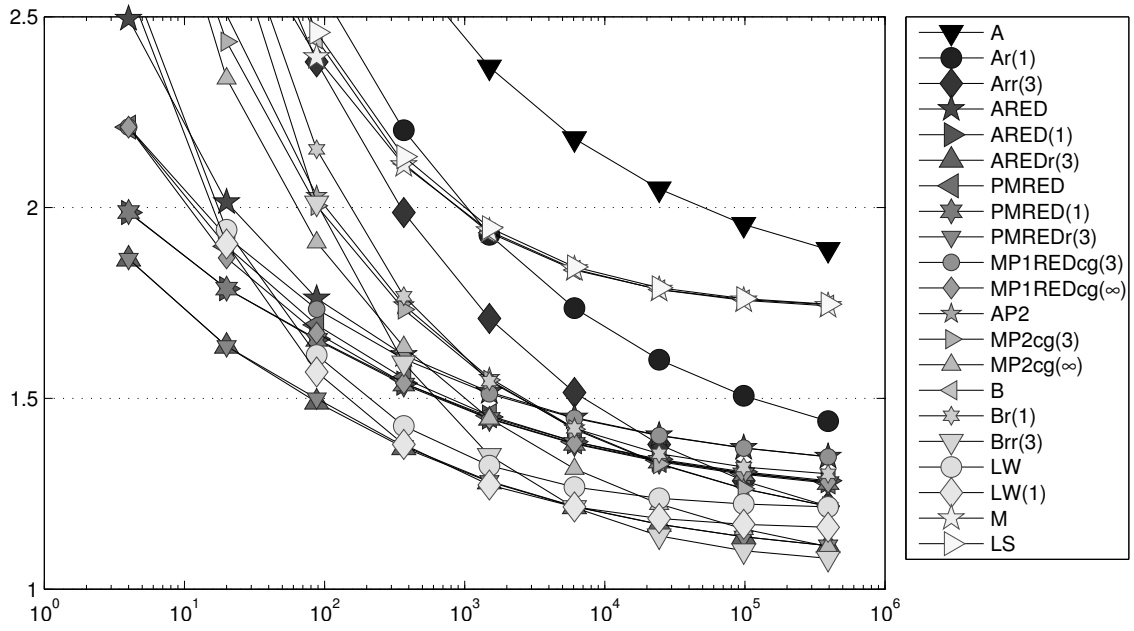

Figure 4.23: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.5.4.

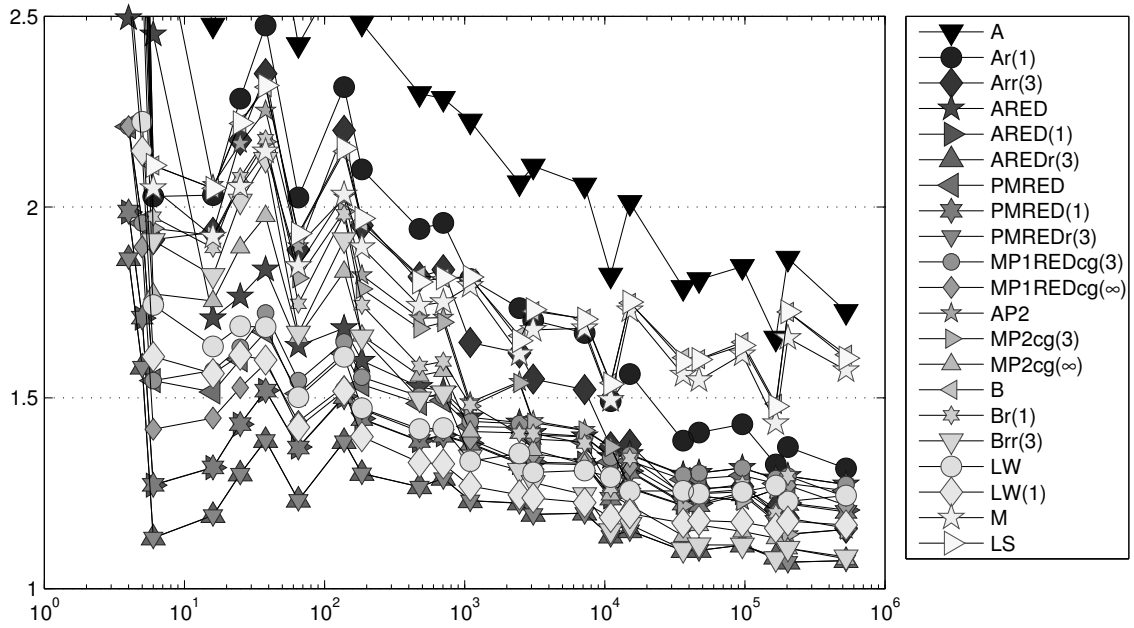

Figure 4.24: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.5.4.

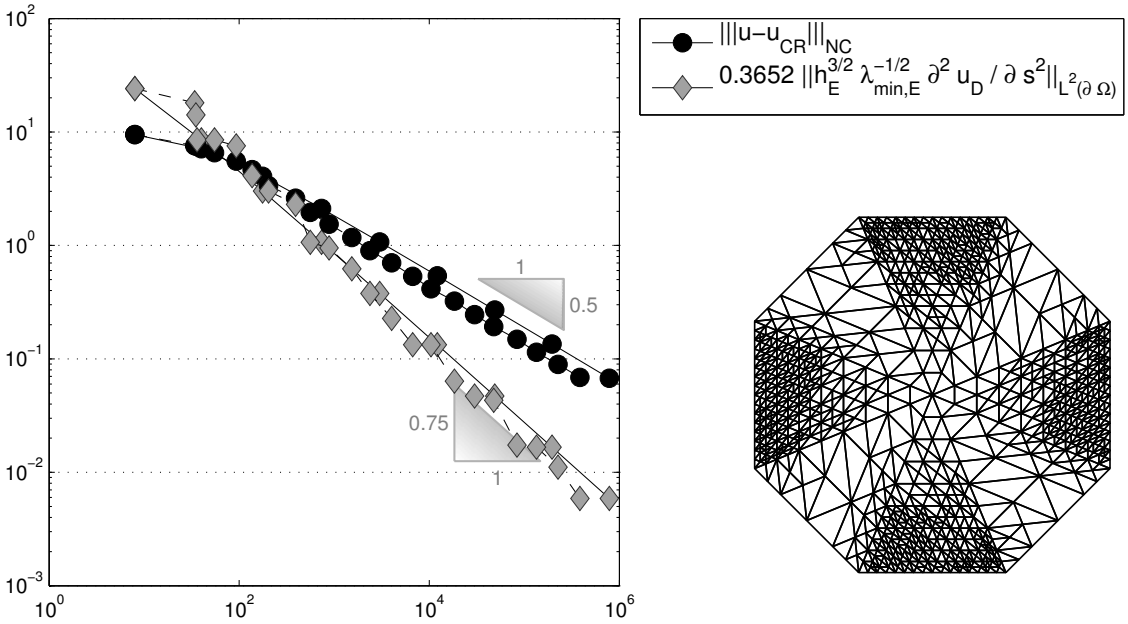

Figure 4.25: Convergence history for the energy error $|||e|||_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \lambda_{\min,\mathcal{E}}^{-1/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.5.5 with respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$. The right image shows the adaptive mesh on level 12.

diffusion coefficients, but still better than the error estimator η_{ARED} related to the initial value of the pcg scheme that assumes efficiency indices around 1.3. The situation is even more dramatic for $\eta_{\mathrm{MP1REDcg}(3)}$. It is based on the same initial value and the efficiency hardly shows any improvement after three cg iterations. The optimal error estimator $\eta_{\mathrm{MP1REDcg}(\infty)}$ leads to slightly improved efficiency indices around 1.2.

4.6 Possible Modifications for Nonpolygonal Domains

Triangulations into triangles are not able to approximate domains with curved boundaries exactly and thus lead to additional error quantities. This section discusses a 2D example that avoids curved finite elements.

As an example, consider the benchmark problem from Ainsworth (2004) on the sector domain

$$\Omega = \left\{ x = (r\cos\varphi, r\sin\varphi) \,\middle|\, 0 < \varphi < 3\pi/2,\ 0 < r < 1 \right\}$$

with a reentrant corner at $(0,0)$ and the exact solution $u(r,\varphi) = (r^{2/3} - r^2)\sin(2\varphi/3)$.

4.6.1 Conforming $\mathcal{P}_1$-FEM

To fully cover the domain, the triangulation is extended with reflection of the boundary triangles as shown in Figure 4.28. The resulting triangulation $\widehat{\mathcal{T}}$ satisfies $\Omega \subset \bigcup \widehat{\mathcal{T}}$ where the extended source function $f(\varphi) = 32\sin(2\varphi/3)/9$ is well defined. The discrete solution

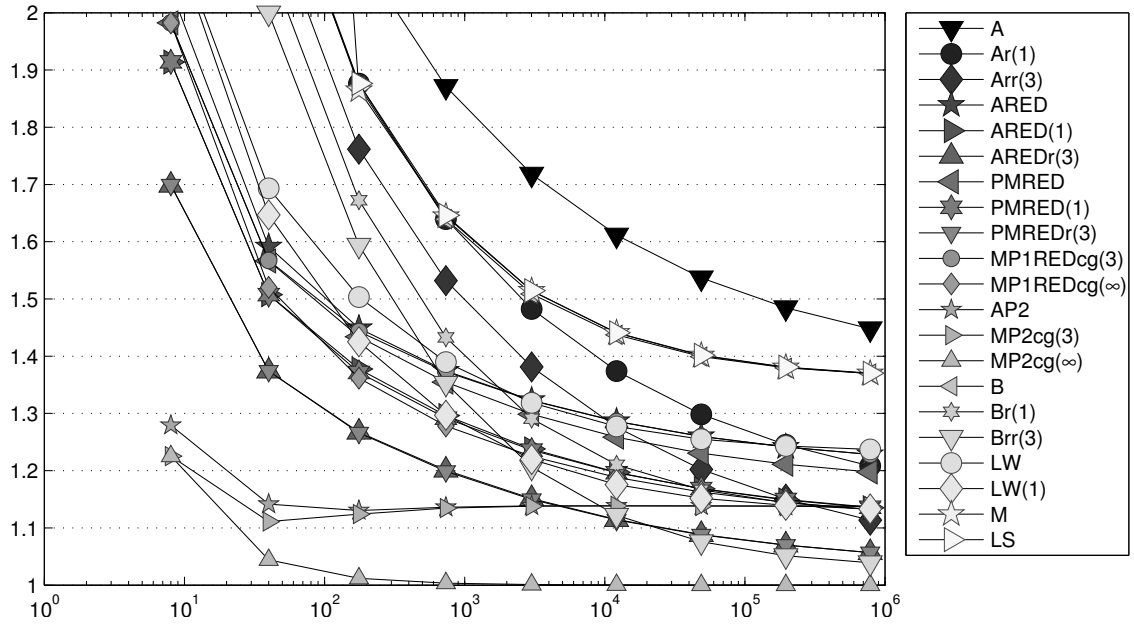

Figure 4.26: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.5.5.

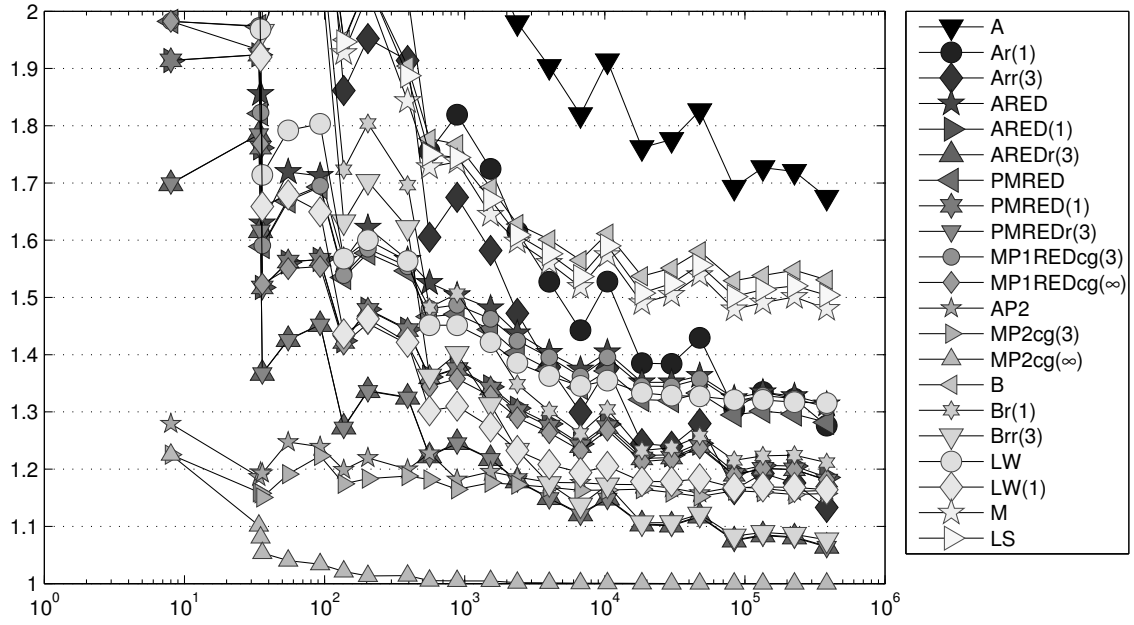

Figure 4.27: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.5.5.

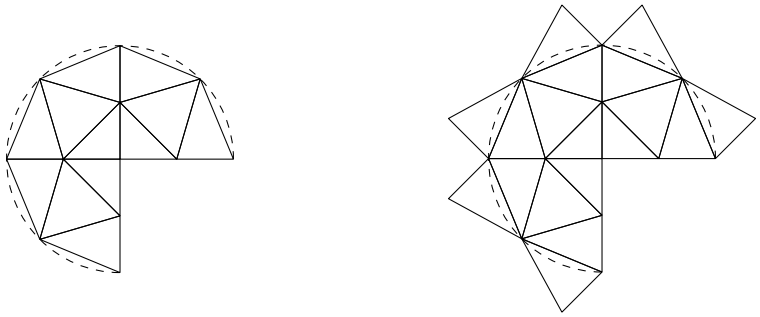

Figure 4.28: Triangulation $\mathcal{T}$ (left, solid) and extended triangulation $\widehat{\mathcal{T}}$ (right, solid) with $\bigcup\mathcal{T} \subseteq \Omega \subseteq \bigcup\widehat{\mathcal{T}}$ for the sector domain Ω (dashed).

u_h (computed with $u_D = 0$ along $\partial(\bigcup\mathcal{T})$) is extended by zero outside of $\bigcup\mathcal{T}$ such that $u_h \in H^1_0(\Omega)$. This also extends its fluxes $\sigma_h := \nabla u_h$ by zero outside of $\bigcup\mathcal{T}$.

The new triangles involve only Dirichlet nodes and allow the Braess or Luce-Wohlmuth design of an equilibration q_{B} or q_{LW} from Section 3.2 on the extended triangulation, possibly with an additional postprocessing $\gamma_h \in H^1(\bigcup\widehat{\mathcal{T}})$. Then, the error for the conforming finite element method is bounded by

$$|||u - u_h||| = |||\mathrm{Res}|||_\star \leqslant \left\| h_{\widehat{\mathcal{T}}}(f + \operatorname{div}\widehat{q}) \right\|_{L^2(\bigcup\widehat{\mathcal{T}})} / j_{1,1} + \|\widehat{q} - \sigma_h - \operatorname{Curl}\gamma_h\|_{L^2(\Omega)}\,.$$

The integration of $\widehat{q} - \sigma_h - \operatorname{Curl}\gamma_h$ over the non-polygonal domain Ω separates into an exact integration over triangles in $\mathcal{T}$ and an integration over intersections $T \cap \Omega$ of triangles $T \in \widehat{\mathcal{T}}\backslash\mathcal{T}$. The latter integration employs polar coordinates and Gauss quadrature with at least 100 quadrature points. The exact energy error is computed with the usual quadrature on $\mathcal{T}$ and polar coordinates and Gauss quadrature on the domain remainder $\Omega\backslash\bigcup\mathcal{T}$.

For the adaptive mesh refinement, the refinement indicators $\eta(T)^2$ from (4.7) are replaced by

$$\eta(T)^2 + 4\,\mathrm{width}(\widehat{T} \cap \Omega)^2/\pi^2 \;\|f\|^2_{L^2(\widehat{T}\cap\Omega)} \quad \text{for } T \in \mathcal{T} \text{ with a reflection } \widehat{T} \in \widehat{\mathcal{T}}\backslash\mathcal{T}.$$

The additional quantity bounds the integral

$$\int_{\Omega\backslash\bigcup\mathcal{T}} fv\,\mathrm{d}x \leqslant \sum_{T\in\widehat{\mathcal{T}}\backslash\mathcal{T}} 2\,\mathrm{width}(\widehat{T}\cap\Omega)/\pi \;\|f\|_{L^2(\widehat{T}\cap\Omega)}\,\|\nabla v\|_{L^2(\widehat{T}\cap\Omega)}$$

by 1D Friedrichs inequalities along lines orthogonal to E. This causes the refinement along the curved boundary depicted in the adaptive mesh from Figure 4.29. The convergence history of the energy error in Figure 4.29 proves that the extensions do not harm the optimal convergence speed of the adaptive algorithm in the long run.

Figures 4.30 and 4.31 display the efficiency indices of the error estimators. For uniform mesh refinement, the efficiency indices become slightly worse on finer meshes. This may be caused by the extension on the reflected triangles. The Braess error estimator, which is extended to the complete reflected triangles, is more affected than the Luce-

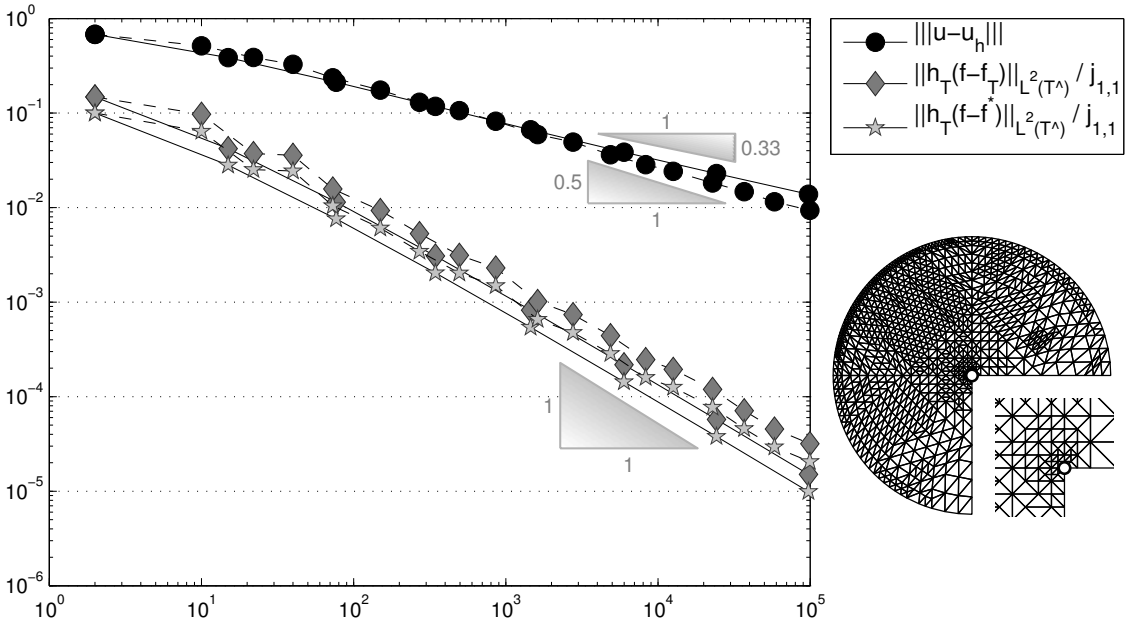

Figure 4.29: Convergence history for the energy error $|||u - u_h|||$ and the higher-order terms in the guaranteed upper bounds of η_{B} and η_{LW} on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.6.1 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 8.

Wohlmuth error estimator, which employs the dual mesh and thus has more flexibility on the reflected triangles. For adaptive mesh refinement, the efficiency indices stay below 1.5 on all meshes for all error estimators and the postprocessing $\eta_{\mathrm{LW}(1)}$ yields efficiency indices around 1.15. Opposite to all other examples, the gap between the truncated postprocessing and the optimal postprocessing appears slightly larger and, for the first time, $\eta_{\mathrm{Brr}(3)}$ is less efficient than $\eta_{\mathrm{Br}(\infty)}$. Somehow, the boundary extension pollutes the convergence speed of the pcg scheme behind the postprocessing.

4.6.2 Nonconforming CR-FEM

Also the nonconforming solution u_{CR} is extended by zero outside of $\bigcup \mathcal{T}$ such that $u_{\mathrm{CR}} = 0$ along $\partial\Omega \backslash \bigcup \mathcal{T}$. Similarly, the design of any v from Subsection 4.4.2 on $\mathcal{T}$ or $\mathrm{red}(\mathcal{T})$ can be extended $H^1(\Omega)$-conformly by $v_{\mathrm{xyz}} = 0$ on $\Omega \backslash \bigcup \mathcal{T}$. Since the normal fluxes of q are zero along $\partial \bigcup \mathcal{T}$ for any equilibration design from Chapter 3 for the residual $\mathrm{Res}_{\mathrm{NC}}$, also q can be extended $H(\mathrm{div}, \Omega)$-conformly by $q_{\mathrm{xyz}} = 0$ on $\Omega \backslash \bigcup \mathcal{T}$. This leads to the guaranteed upper bound

$$|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}^2 \leqslant \widehat{\eta}_0^2 + \mu_{\mathrm{xyz}}^2.$$

with the modified first contribution

$$\widehat{\eta}_0^2 := \eta_0^2 + \sum_{E \in \mathcal{E}(\partial \bigcup \mathcal{T})} \left(\frac{\lambda_{\min,E}^{-1/2} \, |\widehat{\omega}_E|^{1/2}}{|E|} \, |\mathrm{Res}(\psi_E)| + \frac{2\mathrm{width}(\widehat{\omega}_E)}{\pi} \left\| \lambda_{\min,E}^{-1/2} f \right\|_{L^2(\widehat{\omega}_E)} \right)^2 .$$

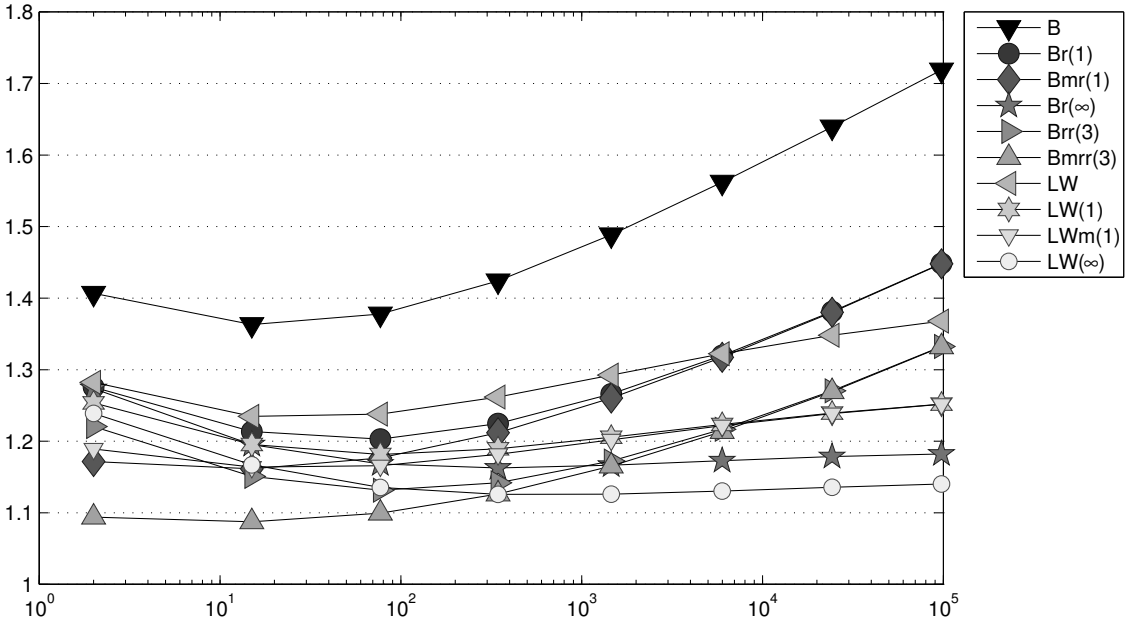

Figure 4.30: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.5.4.

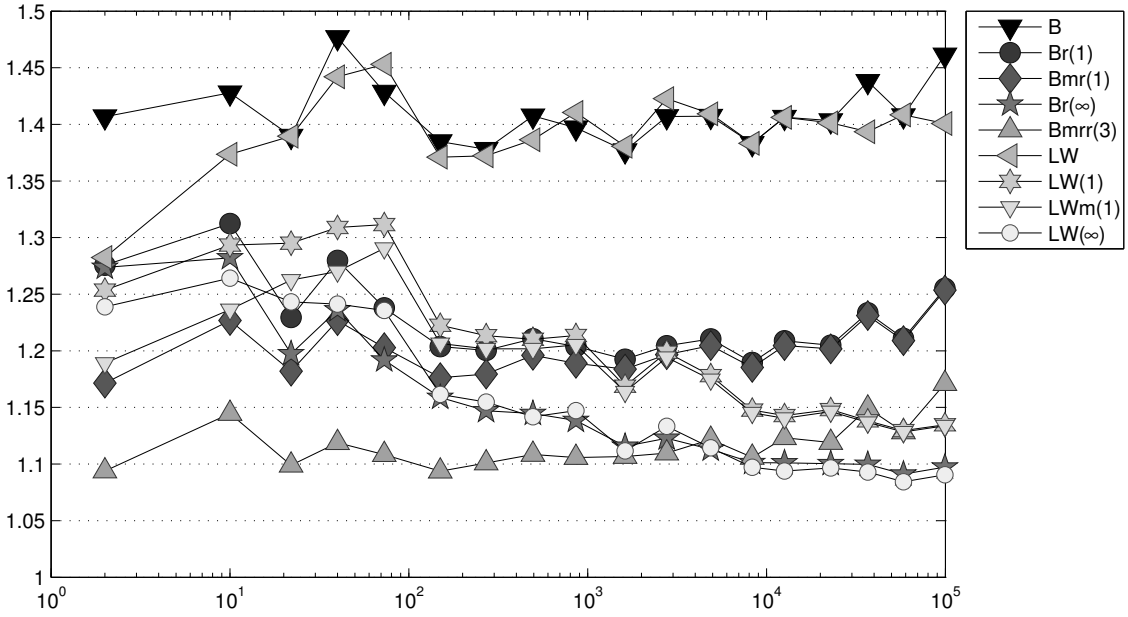

Figure 4.31: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_h|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.6.1.

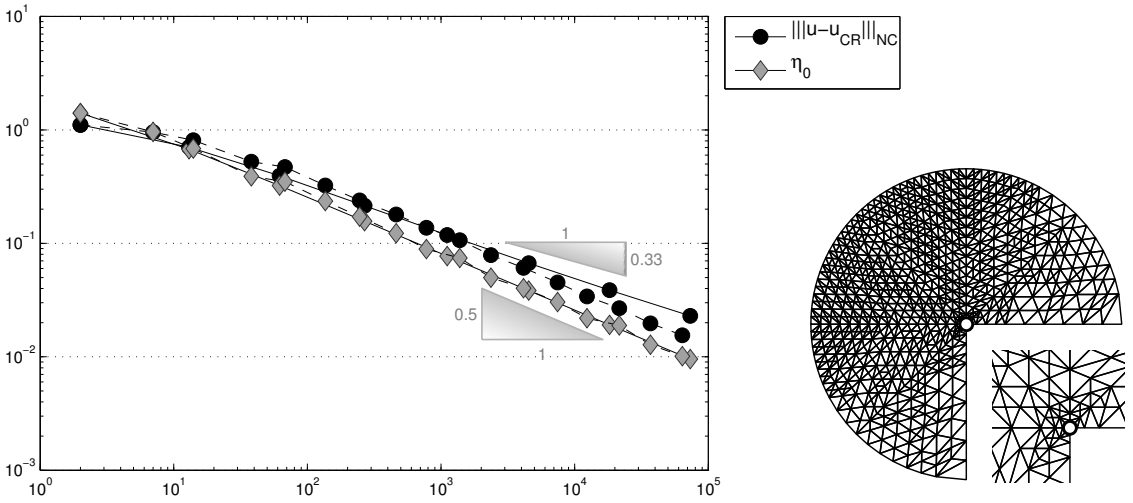

Figure 4.32: Convergence history for the energy error $|||e|||_{\mathrm{NC}}$ and $\widehat{\eta}_0$ (labelled as η_0) on uniform (solid line) and adaptive (dashed line) meshes in Subsection 4.6.2 with respect to the number of degrees of freedom $|\mathcal{E}(\Omega)|$. The right image shows the adaptive mesh on level 10.

The additional contributions along the boundary edges $\mathcal{E}(\partial \bigcup \mathcal{T})$ of the triangulation estimate the influence of α from the Helmholtz decomposition Theorem 4.4.1 on $\Omega \backslash \bigcup \mathcal{T}$. The circular segments $\widehat{\omega}_E$ are enclosed by the circle line $\partial\Omega$ and the edge $E \in \mathcal{E}(\partial \bigcup \mathcal{T})$. Note that the integrals in $\mathrm{Res}(\psi_E)$ are evaluated only on $\bigcup \mathcal{T}$. There are two new terms in the definition of $\widehat{\eta}_0$ compared to η_0 from (4.12). They stem from additional integrals, which arise in the proof of Theorem 4.4.1.(c), due to $\alpha \neq 0$ along the boundary of $\bigcup \mathcal{T} \backslash \partial\Omega$, i.e.,

$$\sum_{E \in \mathcal{E}(\partial \bigcup \mathcal{T})} \int_{\widehat{\omega}_E} f\alpha \,\mathrm{d}x + \fint_E \alpha \,\mathrm{d}s \,\mathrm{Res}(\psi_E).$$

The first integral is estimated by 1D Friedrichs inequalities along lines orthogonal to E, i.e.,

$$\int_{\widehat{\omega}_E} f\alpha \,\mathrm{d}x \leqslant \frac{2\mathrm{width}(\widehat{\omega}_E)}{\pi} \|f\|_{L^2(\widehat{\omega}_E)} \|\nabla\alpha\|_{L^2(\widehat{\omega}_E)}.$$

The estimation of the second integral employs the 1D fundamental theorem of calculus along $\partial\Omega \cap \partial\widehat{\omega}_E$ in outer normal direction ν_E of E and a Cauchy inequality, i.e.,

$$\begin{aligned}\fint_E \alpha \,\mathrm{d}s \,\mathrm{Res}(\psi_E) &= \frac{1}{|E|} \int_E \int_0^{\mathrm{dist}(x, \partial\Omega \cap (x + \nu_E \mathbb{R}))} \nabla\alpha(x + t\nu_E) \cdot \nu_E \,\mathrm{d}t \,\mathrm{d}s_x \,\mathrm{Res}(\psi_E) \\ &\leqslant \frac{1}{|E|} \int_{\widehat{\omega}_E} |\nabla\alpha| \,\mathrm{d}x \,|\mathrm{Res}(\psi_E)| \leqslant \frac{|\widehat{\omega}_E|^{1/2}}{|E|} |\mathrm{Res}(\psi_E)| \,\|\nabla\alpha\|_{L^2(\widehat{\omega}_E)}.\end{aligned}$$

Figure 4.32 shows the convergence history for the energy error and $\widehat{\eta}_0$. The convergence rate for adaptive mesh refinement is optimal and $\widehat{\eta}_0$ also converges with convergence rate

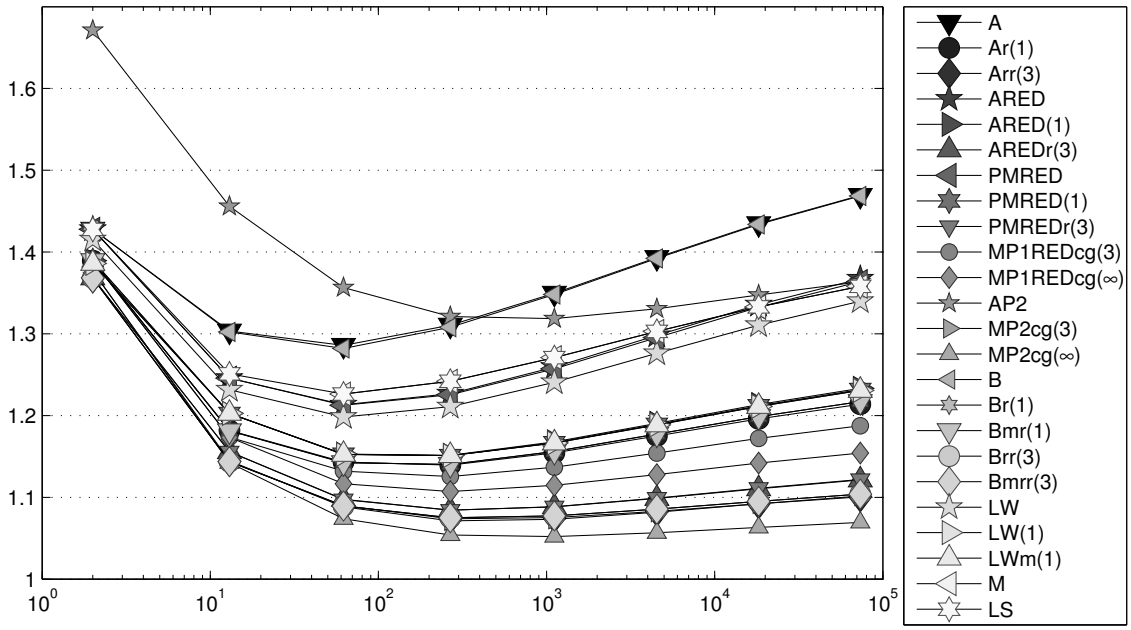

Figure 4.33: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 4.6.2.

1/2. The adaptive mesh refinement concentrates on the reentrant corner.

The efficiency indices displayed in Figures 4.33 and 4.34 are as good as in the other numerical examples.

4.7 Error Analysis for Raviart-Thomas Mixed FEM

The Raviart-Thomas mixed FEM from Subsection 2.3.3 for inhomogeneous Neumann data $g \in L^2(\Gamma_N)$ and its Fortin interpolation $q_{N,RT} = \sum_{E\in\mathcal{E}(\Gamma_N)}(\int_E g\,\mathrm{d}s)\vartheta_E$ (compare with Definition 2.2.16) seeks $q_{\mathrm{RT}} \in q_{N,RT} + H_N(\mathrm{div},\Omega)$ and $u_0 \in \mathcal{P}_0(\mathcal{T})$ with

$$\int_\Omega \mathbb{S}^{-1}q_{\mathrm{RT}}\cdot r_{\mathrm{RT}}\,\mathrm{d}x + \int_\Omega \mathrm{div}\, r_{\mathrm{RT}}u_0\,\mathrm{d}x = 0 \qquad \text{for all } r_{\mathrm{RT}}\in Q(\mathcal{T}),$$
$$\int_\Omega \mathrm{div}\, q_{\mathrm{RT}}v_0\,\mathrm{d}x = \int_\Omega fv_0\,\mathrm{d}s \qquad \text{for all } v_0\in\mathcal{P}_0(\mathcal{T}).$$

The Raviart-Thomas mixed FEM is closely related to the nonconforming Crouzeix-Raviart FEM and the error analysis exploits this relation similar to (Vohralík, 2007; Ainsworth, 2007/08; Braess, 2009). First, consider the function

$$u^\star := u_{\mathrm{CR}} - f_{\mathcal{T}}\,|x-\mathrm{mid}(\mathcal{T})|^2/(2n) \in \mathcal{P}_2(\mathcal{T}). \tag{4.21}$$

The following theorem states two results. The first one considers $u^\star$ as an nonconforming piecewise quadratic approximation of u that allows favourable guaranteed error control. The second result shows that $\mathbb{S}\nabla_{\mathrm{NC}}u^\star$ is superclose (or identical for constant data) to q_{RT}.

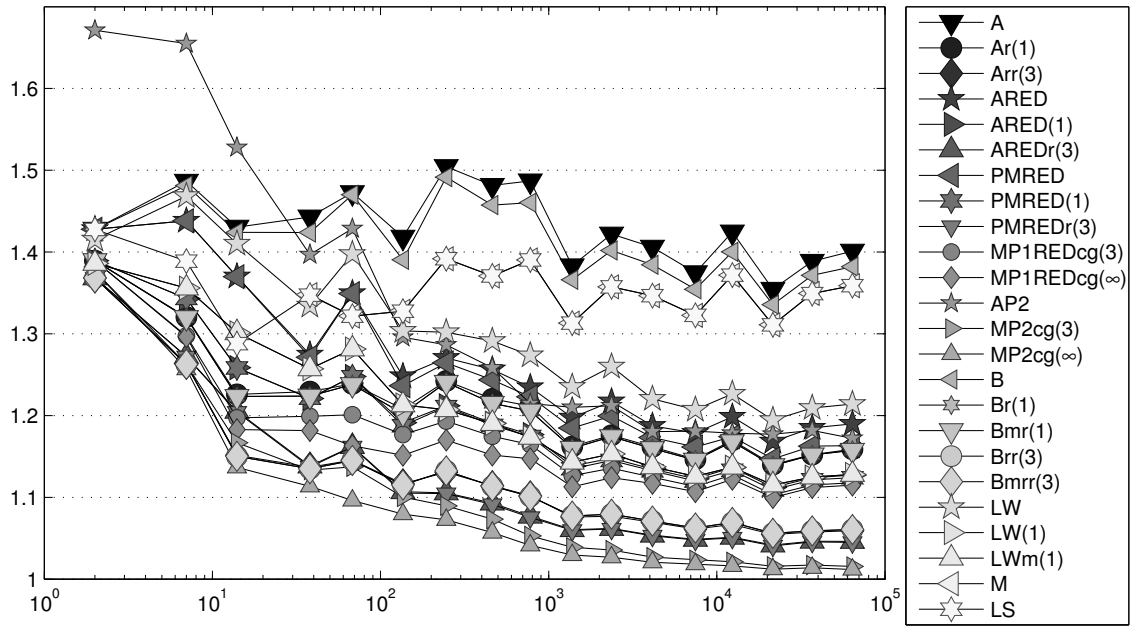

Figure 4.34: History of efficiency indices $\eta_{\mathrm{xyz}} / |||e|||_{\mathrm{NC}}$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 4.6.2.

Theorem 4.7.1. *(a) For simply connected domains* Ω *and* $u^\star$ *from* (4.21)*, it holds*

$$|||u - u^\star|||_{\mathrm{NC}}^2 \leqslant \Big(C_{\mathrm{P}}(\mathcal{T}) \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2} f, \mathcal{T}) + C_{\mathrm{N}}(\mathcal{E}(\Gamma_N)) \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2} g, \mathcal{E}(\Gamma_N)) \Big)^2 + \min_{\substack{v \in H^1(\Omega) \\ v = u_D \text{ on } \Gamma_D}} |||u^\star - v|||_{\mathrm{NC}}^2 .$$

(b) The function $u^\star$ *from* (4.21) *and the Raviart-Thomas mixed FEM solution* q_{RT} *are superclose in the sense*

$$\left\| \mathbb{S}^{-1/2} (q_{\mathrm{RT}} - \mathbb{S} \nabla_{\mathrm{NC}} u^\star) \right\|_{L^2(\Omega)} \leqslant C_{\mathrm{P}}(\mathcal{T}) \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2} f, \mathcal{T}) + C_{\mathrm{N}}(\mathcal{E}(\Gamma_N)) \operatorname{osc}(\lambda_{\min,\mathcal{T}}^{-1/2} g, \mathcal{E}(\Gamma_N)).$$

Proof. The proof of (a) employs the Helmholtz decomposition (Theorem 2.1.12)

$$\nabla_{\mathrm{NC}}(u - u^\star) = \nabla \alpha + \mathbb{S}^{-1} \operatorname{Curl} \beta$$

for some $\alpha \in H_D^1(\Omega)$ and $\beta \in H^1(\Omega; \mathbb{R}^s)$ with $\operatorname{Curl} \beta \cdot \nu = 0$ along Γ_N. Analogously to Theorem 4.4.1.(b), it holds

$$\left\| \mathbb{S}^{-1/2} \operatorname{Curl} \beta \right\|_{L^2(\Omega)} = \min_{\substack{v \in H^1(\Omega) \\ v = u_D \text{ on } \Gamma_D}} |||u^\star - v|||_{\mathrm{NC}}$$

and it remains to estimate $|||\alpha||| \leqslant C_{\mathrm{P}}(\mathcal{T})\operatorname{osc}(f,\mathcal{T})$ which is similar to the estimate in Theorem 4.4.1.(c). The definition of $u^\star$ in (4.21), the properties of the nonconforming interpolation (4.13) and an integration by parts yield

$$\begin{aligned}
|||\alpha||| &= \int_\Omega \mathbb{S}\nabla_{\mathrm{NC}}(u-u^\star)\cdot\nabla\alpha \\
&= F(v) - a_{\mathrm{NC}}(u_{\mathrm{CR}},\alpha) + \frac{1}{n}\int_\Omega f_{\mathcal{T}}(x-\operatorname{mid}(\mathcal{T}))\cdot\nabla\alpha\,\mathrm{d}x \\
&= F(v) - a_{\mathrm{NC}}(u_{\mathrm{CR}},\alpha_{\mathrm{CR}}) + \frac{1}{n}\sum_{T\in\mathcal{T}} f_{\mathcal{T}}\int_T (x-\operatorname{mid}(T))\cdot\nabla(\alpha-\alpha_{\mathrm{CR}})\,\mathrm{d}x \\
&= \int_\Omega (f-f_{\mathcal{T}})(\alpha-\alpha_{\mathrm{CR}})\,\mathrm{d}x + \int_{\Gamma_N}(g-g_{\mathcal{E}})(\alpha-\alpha_{\mathrm{CR}})\,\mathrm{d}s \\
&\leqslant \left(C_{\mathrm{P}}(\mathcal{T})\operatorname{osc}(f,\mathcal{T}) + C_{\mathrm{N}}(\mathcal{E}(\Gamma_N))\operatorname{osc}(g,\mathcal{E}(\Gamma_N))\right)|||\alpha-\alpha_{\mathrm{CR}}|||_{\mathrm{NC}}\,.
\end{aligned}$$

The stability of the nonconforming interpolation $|||\alpha-\alpha_{\mathrm{CR}}|||_{\mathrm{NC}} \leqslant |||\alpha|||$ (which is a consequence of (4.13)) concludes the proof of (a).

The proof of (b) employs the equivalence

$$q_{\mathrm{RT}} := \mathbb{S}\nabla_{\mathrm{NC}}\widehat{u}_{\mathrm{CR}} - \frac{f_{\mathcal{T}}}{n}(\cdot-\operatorname{mid}(\mathcal{T})) \tag{4.22}$$

from Bahriawati and Carstensen (2005, Theorem 7.1) with the Crouzeix-Raviart solution $\widehat{u}_{\mathrm{CR}}$ for the right-hand side data $f_{\mathcal{T}}$ and $g_{\mathcal{E}}$ instead of f and g. Then,

$$\begin{aligned}
\left\|\mathbb{S}^{-1/2}(q_{\mathrm{RT}}-\mathbb{S}\nabla_{\mathrm{NC}}u^\star)\right\|^2_{L^2(\Omega)} &= |||\widehat{u}_{\mathrm{CR}}-u_{\mathrm{CR}}|||^2_{\mathrm{NC}} \\
&= \int_\Omega \mathbb{S}\nabla_{\mathrm{NC}}(\widehat{u}_{\mathrm{CR}}-u_{\mathrm{CR}})\cdot\nabla_{\mathrm{NC}}(\widehat{u}_{\mathrm{CR}}-u_{\mathrm{CR}}) \\
&= \int_\Omega (f-f_{\mathcal{T}})(\widehat{u}_{\mathrm{CR}}-u_{\mathrm{CR}})\,\mathrm{d}x + \int_{\Gamma_N}(g-g_{\mathcal{E}})(\widehat{u}_{\mathrm{CR}}-u_{\mathrm{CR}})\,\mathrm{d}s.
\end{aligned}$$

These integrals lead to the asserted oscillations and conclude the proof. □

5 Error Analysis for the Stokes Problem

The Stokes equations result from a simplification of the Navier-Stokes equations and describe the motion of incompressible Newtonian fluids like water.

5.1 Setting, Deviatoric Stress Tensor and Inf-Sup Condition

Given a right-hand side $f \in L^2(\Omega;\mathbb{R}^n)$ and Dirichlet data $u_D \in H^1(\Omega;\mathbb{R}^n)$, the Stokes problem seeks a pressure $p \in L^2_0(\Omega) \coloneqq \{v \in L^2(\Omega) \,\big|\, \int_\Omega v \,\mathrm{d}x = 0\}$ and a velocity field $u \in H^1(\Omega;\mathbb{R}^n)$ with

$$-\Delta u - \nabla p = f, \qquad \operatorname{div} u = 0 \text{ in } \Omega \quad \text{and} \quad u = u_D \text{ on } \partial\Omega. \tag{5.1}$$

Moreover, the Gauss theorem reveals that u_D has to obey the compatibility condition

$$0 = \int_\Omega \operatorname{div} u \,\mathrm{d}x = \int_{\partial\Omega} u_D \cdot \nu \,\mathrm{d}s.$$

The stress tensor $\sigma \coloneqq \mathrm{D}u + p\mathbb{I} \in L^2(\Omega;\mathbb{R}^{n\times n})$ satisfies the equilibrium equation $\operatorname{div}\sigma + f = 0$. The weak formulation of the Stokes problem involves the linear form $F(v) \coloneqq \int_\Omega f \cdot v \,\mathrm{d}x$ for $v \in H^1(\Omega;\mathbb{R}^n)$ and the bilinear forms

$$\begin{aligned} a &: H^1(\Omega;\mathbb{R}^n) \times H^1(\Omega;\mathbb{R}^n) \to \mathbb{R}, & a(u,v) &\coloneqq \int_\Omega \mathrm{D}u : \mathrm{D}v \,\mathrm{d}x, \\ b &: L^2(\Omega) \times H^1(\Omega;\mathbb{R}^n) \to \mathbb{R}, & b(q,v) &\coloneqq \int_\Omega q \operatorname{div} v \,\mathrm{d}x. \end{aligned}$$

Then, a weak solution (u,p) of (5.1) is characterised by the variational equations

$$\begin{aligned} a(u,v) + b(p,v) &= F(v) && \text{for all } v \in H^1_0(\Omega;\mathbb{R}^n), \\ b(q,u) &= 0 && \text{for all } q \in L^2_0(\Omega). \end{aligned} \tag{5.2}$$

In the subsequent error analysis the stress tensor σ is (pointwise almost everywhere) decomposed in two components. One is the hydrostatic or volumetric stress tensor $\operatorname{tr}(\sigma)/n\,\mathbb{I} \in L^2(\Omega;\mathbb{R}^{n\times n})$ and the remaining part defines the deviatoric component

$$\operatorname{dev}(\sigma) \coloneqq \sigma - \operatorname{tr}(\sigma)/n\,\mathbb{I} \ \in L^2(\Omega;\mathbb{R}^{n\times n}).$$

This decomposition is orthogonal in the sense that

$$\operatorname{dev}(\sigma(x)) : (\operatorname{tr}(\sigma(x))/n\,\mathbb{I}) = 0 \quad \text{for a.e. } x \in \Omega.$$

This leads to the L^2 orthogonality

$$\|\sigma\|^2_{L^2(\Omega)} = \|\mathrm{dev}(\sigma)\|^2_{L^2(\Omega)} + \|\mathrm{tr}(\sigma)\|^2_{L^2(\Omega)}/n. \tag{5.3}$$

Notice also that $\mathbb{I} : \mathrm{D}v = \mathrm{div}\, v$ and so

$$F(v) = \int_\Omega \sigma : \mathrm{D}v \,\mathrm{d}x = \int_\Omega \mathrm{D}u : \mathrm{D}v \,\mathrm{d}x \tag{5.4}$$

for all v in the space of divergence-free functions

$$Z := \{v \in H^1_0(\Omega;\mathbb{R}^n) \,\big|\, \mathrm{div}\, v = 0\}.$$

The error analysis below involves the constant c_0 in the inf-sup condition

$$0 < c_0 := \inf_{q\in L^2_0(\Omega)\setminus\{0\}} \sup_{v\in H^1_0(\Omega;\mathbb{R}^n)} \frac{b(q,v)}{\|\mathrm{D}v\|_{L^2(\Omega)}\,\|q\|_{L^2(\Omega)}}. \tag{5.5}$$

It depends only on the domain Ω and equals the smallest eigenvalue of some general eigenvalue problem. Theoretical background and guaranteed, but also rather pessimistic, lower bounds can be found in (Stoyan, 1999). For the error analysis, the following Lemma yields a helpful decomposition of the stress tensor from Dari et al. (1995) and Ainsworth and Dörfler (2005).

Lemma 5.1.1. *Let Ω be a bounded Lipschitz domain.*

(a) Every $p \in L^2(\Omega;\mathbb{R}^{n\times n})$ can be decomposed into

$$p = \mathrm{D}z + y$$

with some divergence-free function $z \in Z$ with

$$\int_\Omega \mathrm{D}z : \mathrm{D}v \,\mathrm{d}x = \int_\Omega p : \mathrm{D}v \,\mathrm{d}x \quad \textit{for all } v \in Z, \tag{5.6}$$

and the remainder

$$y \in Y := \left\{ y \in L^2(\Omega;\mathbb{R}^{n\times n}) \,\big|\, \int_\Omega y : \mathrm{D}v \,\mathrm{d}x = 0 \quad \textit{for all } v \in Z \right\}.$$

(b) For each $y \in Y$, there exists some $w \in L^2_0(\Omega)$ with

$$\int_\Omega y : \mathrm{D}v \,\mathrm{d}x = \int_\Omega w\, \mathrm{div}\, v \,\mathrm{d}x \quad \textit{for all } v \in H^1_0(\Omega;\mathbb{R}^n) \tag{5.7}$$

and

$$\|w\|_{L^2(\Omega)} \leqslant 1/c_0 \;\|y\|_{L^2(\Omega)}.$$

Proof. For a proof of (a), see Dari et al. (1995, Lemma 3.2) or Ainsworth and Dörfler (2005, Subsection 3.2). The function $z \in Z$ is the unique solution of the problem (5.6) and the

remainder obviously is in Y. A proof of (b) can be found in Ainsworth and Dörfler (2005, Lemma 2) and is based on the theory of Galdi (1994, Chapter III.1). □

5.2 Error Analysis for Conforming Finite Element Methods

This section deals with conforming approximations to the Stokes problem and includes the mini finite element approach. However, the general results also hold for the Taylor-Hood finite element method (or any other conforming finite element method).

5.2.1 The Mini FEM for the Stokes Problem

The mini finite element space consists of functions in $V(\mathcal{T}) = \mathcal{P}_1(\mathcal{T}) \cap V$ and additional element volume bubbles $b_T \coloneqq \prod_{z\in\mathcal{N}(T)} \varphi_z \in \mathcal{P}_3(T)$ on each triangle $T \in \mathcal{T}$, i.e.,

$$\mathcal{B}(\mathcal{T}) \coloneqq \operatorname{span}\{b_T \,|\, T \in \mathcal{T}\}, \qquad \operatorname{Mini}(\mathcal{T}) \coloneqq V(\mathcal{T};\mathbb{R}^n) \oplus \mathcal{B}(\mathcal{T};\mathbb{R}^n).$$

The nodal interpolation operator $u_{D,h} \coloneqq \sum_{z\in\mathcal{N}(\partial\Omega)} u(z)\varphi_z$ interpolates the inhomogeneous Dirichlet data and the mini finite element methods seeks $u_M \in u_{D,h} + \operatorname{Mini}(\mathcal{T})$, $p_h \in V(\mathcal{T})$ and $\lambda \in \mathbb{R}$ with

$$\begin{aligned} a(u_M, v_M) + b(p_h, v_M) &= F(v_M), \\ b(q_h, u_M) + \lambda \int_\Omega q_h \,\mathrm{d}x &= 0, \\ \int_\Omega p_h \,\mathrm{d}x &= 0 \qquad \text{for all } v_M \in \operatorname{Mini}(\mathcal{T}) \text{ and } q_h \in V(\mathcal{T}). \end{aligned} \tag{5.8}$$

The number λ is a Lagrange multiplier for the constraint $\int_\Omega p_h \,\mathrm{d}x = 0$.

Listing 5.1 displays the `MATLAB` code for the mini finite element solver. The array `A4e` contains the local stiffness matrices of the nodal basis functions and the bubble functions for the first component of u_h (the entries for the second component are identical and copied in Line 22). The submatrix `A4e(1:3,1:3,k)` equals the local stiffness matrix from Section 2.3.5.4 for the k-th element of the triangulation. Note that the definition of b_T and Lemma 2.2.18 lead to

$$\int_T \nabla b_T \cdot \nabla \varphi_z \,\mathrm{d}x = 0 \quad \text{for any } z \in \mathcal{N}(T). \tag{5.9}$$

This results in `A4e(1:3,4,k) = 0` and the remaining contributions `A4e(4,4,k)` of the bubble functions in Line 10 can be computed analytically by Lemma 2.2.18, i.e.,

$$\int_T \nabla b_T \cdot \nabla b_T \,\mathrm{d}x = \frac{|T|}{180} \sum_{z\in\mathcal{N}(T)} |\nabla \varphi_z|_T|^2 .$$

The array `B4e` contains the local matrices for the discretisation of the bilinear form b where the nodal basis functions for the pressure are multiplied with the divergences of the eight basis functions of the velocity (three nodal basis functions and one bubble function

for each component), which are essentially partial derivatives of the nodal basis functions. The loop in Lines 5–17 computes all the local matrix entries elementwise. Lines 18–28 sum up all local matrix entries into the global stiffness matrices `A` and `B` that discretise the two bilinear forms a and b, i.e., That means

$$\texttt{A(j,k)} = \int_\Omega Dv_j : Dv_k \,\mathrm{d}x \quad \text{for } j,k = 1,\dots,2(|\mathcal{N}| + |\mathcal{T}|),$$
$$\texttt{B(j,k)} = \int_\Omega \varphi_j : \operatorname{div} v_k \,\mathrm{d}x \quad \text{for } j = 1,\dots,|\mathcal{N}| \text{ and } k = 1,\dots,2(|\mathcal{N}| + |\mathcal{T}|).$$

The first $|\mathcal{N}|$ basis functions read $v_j = (\varphi_{z_j}, 0)$ for the nodal basis functions $\varphi_j, j = 1, \dots, |\mathcal{N}|$ of the j-th node `c4n(j,:)`. The next $|\mathcal{T}|$ basis functions read $v_{j+|\mathcal{N}|} = (b_{T_j}, 0)$ for the bubble functions b_T for the triangle j-th triangle `n4e(j,:)`. The remaining $|\mathcal{N}| + |\mathcal{T}|$ basis functions do the same for the second component. Lines 29–31 calculate the right-hand side vector by integration of f times basis functions. The integrand for this integration is displayed in Lines 48–54. The remaining lines include the Dirichlet boundary conditions (subtraction of $a(u_{D,h}, \cdot)$ and $b(\cdot, u_{D,h})$ from the right-hand side), the setup of the system matrix and the solve of the linear system in (5.8). Lines 42–45 break the solution vector `x` into the velocity part `u` and the pressure part `p`.

5.2.2 Error Analysis

The theorem of this subsection enables guaranteed upper bounds for the energy norm $|||e||| = \|\mathrm{D}e\|_{L^2(\Omega)}$ of the error $e := u - u_h$ between u and any conforming approximation u_h.

The stress $\sigma_h := \mathrm{D}u_h + p_h\mathbb{I}$ is a discrete approximation of the exact stress σ and fits in the framework of the unified error analysis of Chapter 3 with the residual

$$\mathrm{Res}(v) := \int_\Omega f \cdot v \,\mathrm{d}x - \int_\Omega \sigma_h : \mathrm{D}v \,\mathrm{d}x \quad \text{for } v \in Z. \tag{5.10}$$

Recall that $|||\mathrm{Res}|||_{Z\setminus\{0\}^\star} = \sup_{z\in Z} \mathrm{Res}(z)/|||z|||$ denotes the dual norm of Res with respect to the space Z.

Theorem 5.2.1. *(a) It holds*

$$|||u - u_h|||^2 \leqslant |||\mathrm{Res}|||_{Z^\star}^2 + 1/c_0^2 \,\|\operatorname{div} u_h\|_{L^2(\Omega)}^2 .$$

with c_0 from the inf-sup condition (5.5) *and the residual* Res *from* (5.10).
(b) Efficiency holds in the sense of

$$|||\mathrm{Res}|||_{Z^\star}^2 + 1/c_0^2 \,\|\operatorname{div} u_h\|_{L^2(\Omega)}^2 \leqslant \max\{1, c_0^{-2}/n\} \,|||u - u_h|||^2 .$$

Proof. Proof of (a). Lemma 5.1.1.(a) yields the decomposition of

$$\mathrm{D}e = \mathrm{D}(u - u_h) = \mathrm{D}z + y$$

```
[u,p] = solveMINIStokes(f,u4Db,c4n,n4e,n4sDb,degree_f)
A4e = zeros(4,4,nrElems);
B4e = zeros(8,3,nrElems);
grads4e = zeros(3,2,nrElems);
area4e = computeArea4e(c4n,n4e);
area4n = computeArea4n(c4n,n4e);
for j=1:nrElems
    grads = [ones(1,3);c4n(n4e(j,:),:)']\[zeros(1,2);eye(2)];
    A4e(1:3,1:3,j) = area4e(j)*(grads*grads');
    A4e(4,4,j) = sum(diag(A4e(1:3,1:3,j)))/180;
    B4e([1:3 5:7],1,j) = grads(:)';
    B4e([1:3 5:7],2,j) = grads(:)';
    B4e([1:3 5:7],3,j) = grads(:)';
    B4e([4 8],:,j) = -grads'/20;
    B4e(:,:,j) = B4e(:,:,j)*area4e(j)/3;
    grads4e(:,:,j) = grads;
end
dofs_u = [n4e'; (nrNodes+1:nrNodes+nrElems)];
I = repmat(dofs_u(:),1,size(dofs_u,1))';
J = repmat(dofs_u',1,size(dofs_u,1))';
A = sparse(I(:),J(:),A4e(:));
A = [A sparse(nrNodes+nrElems,nrNodes+nrElems)
     sparse(nrNodes+nrElems,nrNodes+nrElems) A];
dofs_u = [dofs_u; nrNodes+nrElems+dofs_u]';
dofs_p = n4e';
I = repmat(dofs_p(:),1,size(dofs_u,2))';
J = repmat(dofs_u,1,size(dofs_p,1))';
B = sparse(I(:),J(:),B4e(:),nrNodes,2*(nrNodes+nrElems));
integrand = @(n4p,pts,pts_ref)RHS_BasisXf(n4p,pts,pts_ref,f);
b4e = integrate(c4n,n4e,integrand,degree_f+3);
b = accumarray(dofs_u(:),b4e(:),[3*nrNodes+2*nrElems+1 1]);
DbNodes = unique(n4sDb(:));
fixeddofs = [DbNodes' nrNodes+nrElems+DbNodes'];
x = zeros(3*nrNodes+2*nrElems+1,1);
x(fixeddofs)=reshape(u4Db(c4n(DbNodes,:)),[1 2*length(DbNodes)]);
M=[A B' sparse(2*(nrNodes+nrElems),1);...
   B sparse(nrNodes,nrNodes) area4n/3;...
   sparse(1,2*(nrNodes+nrElems)) area4n'/3 0];
b = b - M*x;
dofs=setdiff(1:3*nrNodes+2*nrElems+1,fixeddofs);
x(dofs)=M(dofs,dofs)\b(dofs);
u1 = x(1:nrNodes+nrElems);
u2 = x(nrNodes+nrElems+1:2*(nrNodes+nrElems));
u=[u1 u2];
p=x(2*(nrNodes+nrElems)+1:end-1);
end

function val = RHS_BasisXf(n4p,pts,pts_ref,f)
  x = pts_ref(1); y = pts_ref(2);
  f4pts = f(pts);
  f4pts = f4pts(:,[1 1 1 1 2 2 2 2]);
  basis4pts = repmat([1-x-y x y x*y*(1-x-y)],[size(pts,1) 2]);
  val = f4pts.*basis4pts;
end
```

Listing 5.1: Listing for the mini fem solver `solveMINIStokes`.

into some $z \in Z$ and $y \in Y$ that are orthogonal in the sense that

$$|||e|||^2 = |||z|||^2 + \|y\|^2_{L^2(\Omega)} = \int_\Omega De : Dz \,dx + \int_\Omega De : y \,dx.$$

Since $\mathbb{I} : Dz = \operatorname{div} z = 0$, the first term equals

$$\int_\Omega De : Dz \,dx = \int_\Omega (\sigma - \sigma_h) : Dz \,dx = \operatorname{Res}(z) \leqslant |||\operatorname{Res}|||_{Z^\star} \, |||z||| .$$

It remains to estimate $\int_\Omega De : y \,dx$ and Lemma 5.1.1.(b) shows

$$\begin{aligned}\int_\Omega De : y \,dx = \int_\Omega w \operatorname{div} e \,dx = -\int_\Omega w \operatorname{div} u_h \,dx &\leqslant \|w\|_{L^2(\Omega)} \|\operatorname{div} u_h\|_{L^2(\Omega)} \\ &\leqslant 1/c_0 \, \|\operatorname{div} u_h\|_{L^2(\Omega)} \|y\|_{L^2(\Omega)} .\end{aligned}$$

This concludes the proof of (a).

Proof of (b). The orthogonal decomposition (5.3) for De into its deviatoric and spherical tensor and $\operatorname{div} u = 0$ yield

$$\begin{aligned}|||e|||^2 &= \|\operatorname{dev}(De)\|^2_{L^2(\Omega)} + \|\operatorname{tr}(De)\mathbb{I}\|^2_{L^2(\Omega)}/n \\ &= \|\operatorname{dev}(De)\|^2_{L^2(\Omega)} + \|\operatorname{div}(u_h)\|^2_{L^2(\Omega)}/n. \end{aligned} \tag{5.11}$$

The property (5.4) shows

$$\operatorname{Res}(z) = \int_\Omega (\sigma - \sigma_h) : Dz \,dx = \int_\Omega \operatorname{dev}(\sigma - \sigma_h) : Dz \,dx \qquad \text{for all } z \in Z.$$

Hence, $|||\operatorname{Res}|||_{Z^\star} \leqslant \|\operatorname{dev}(\sigma - \sigma_h)\|_{L^2(\Omega)} = \|\operatorname{dev}(De)\|_{L^2(\Omega)}$. This results in

$$|||\operatorname{Res}|||^2_{Z^\star} + 1/c_0^2 \, \|\operatorname{div} u_h\|^2_{L^2(\Omega)} \leqslant \|\operatorname{dev}(De)\|^2_{L^2(\Omega)} + c_0^{-2} \, \|\operatorname{div} u_h\|^2_{L^2(\Omega)} .$$

The coefficient comparison with (5.11) concludes the proof of (b). □

The factor $\|\operatorname{div} u_h\|_{L^2(\Omega)}/(c_0 \, |||e|||)$, leads to some lower threshold for the efficiency index of any error estimator $\eta \geqslant |||\operatorname{Res}|||_{Z^\star}$, i.e.,

$$\frac{\left(\eta^2 + \|\operatorname{div} u_h\|^2_{L^2(\Omega)}/c_0^2\right)^{1/2}}{|||u - u_h|||} \geqslant \left(\frac{|||\operatorname{Res}|||^2_{Z^\star}}{|||u - u_h|||^2} + \frac{\|\operatorname{div} u_h\|^2_{L^2(\Omega)}}{c_0^2 \, |||u - u_h|||^2}\right)^{1/2} \geqslant \frac{\|\operatorname{div} u_h\|_{L^2(\Omega)}}{c_0 \, |||u - u_h|||}. \tag{5.12}$$

In the numerical experiments of Section 5.4 the threshold (5.12) often assumes values between 1.5 and 2.5. This implies overestimation of at least 50 to 150 percent solely by the fixed part of the guaranteed upper bound. Therefore, sharper guaranteed lower bounds for c_0 might increase the efficiency dramatically, possibly by nonconforming numerical approximations as in Carstensen and Gedicke (2013+).

5.2.3 Treatment of Inhomogeneous Boundary Data

The boundary extension operator of Section 4.2.2 (componentwise) designs some $w_D \in H^1(\Omega;\mathbb{R}^n)$ with $w_D|_{\partial\Omega} = (u - u_{D,h})|_{\partial\Omega}$ and Theorem 5.2.1.(a) shows

$$|||e - w_D|||^2 \leqslant |||\mathrm{Res}|||^2_{Z^\star} + 1/c_0^2 \, \|\mathrm{div}(u_h + w_D)\|^2_{L^2(\Omega)} \,.$$

A triangle inequality and a Young inequality for any $\lambda > 0$ in the divergence term yields

$$\begin{aligned}
|||e||| &\leqslant |||e - w_D||| + |||w_D||| \\
&\leqslant \left(|||\mathrm{Res}|||^2_{Z^\star} + 1/c_0^2 \, \|\mathrm{div}(u_h + w_D)\|^2_{L^2(\Omega)} \right)^{1/2} + |||w_D||| \\
&\leqslant \min_{\lambda>0} \left(|||\mathrm{Res}|||^2_{Z^\star} + \frac{1+\lambda}{c_0^2} \, \|\mathrm{div}\, u_h\|^2_{L^2(\Omega)} + \frac{1+1/\lambda}{c_0^2} \, \|\mathrm{div}\, w_D\|^2_{L^2(\Omega)} \right)^{1/2} + |||w_D||| \,.
\end{aligned}$$

Theorem 4.2.2 shows

$$\|\mathrm{div}\, w_D\|_{L^2(\Omega)} = \|\mathbb{I} : \mathrm{D} w_D\|_{L^2(\Omega)} \leqslant |||w_D||| \leqslant C_{\mathrm{D},1}(\mathcal{E}(\partial\Omega)) \left\| h_{\mathcal{E}}^{3/2} \partial_{\mathcal{E}}^2 u_{D,h}/\partial s^2 \right\|_{L^2(\partial\Omega)}$$

and the optimal λ reads

$$\lambda = C_{\mathrm{D},1}(\mathcal{E}(\partial\Omega)) \left\| h_{\mathcal{E}}^{3/2} \partial_{\mathcal{E}}^2 u_{D,h}/\partial s^2 \right\|_{L^2(\partial\Omega)} / \|\mathrm{div}\, u_h\|_{L^2(\Omega)} \,.$$

Since, in general, $|||w_D|||$ is of higher order compared to $\|\mathrm{div}\, u_h\|_{L^2(\Omega)}$, λ tends to 0 on fine meshes. However, the factor c_0 appears in front of this term and further pollutes the efficiency on coarse meshes.

5.3 Equilibration for the Mini FEM

This section explains how to apply the equilibration designs of Chapter 3 in the scope of the Stokes mini finite element method.

For all $z \in Z$, $q \in H(\mathrm{div}, \Omega; \mathbb{R}^{n\times n})$ and $\gamma \in H^1(\Omega;\mathbb{R}^n)$, $\mathrm{div}\, z = 0$ and an integration by parts show

$$\mathrm{Res}(z) = \int_\Omega (f + \nabla p_h + \mathrm{div}\, q) \cdot z \,\mathrm{d}x + \int_\Omega \mathrm{dev}(q - \mathrm{D}u_h - \mathrm{Curl}\,\gamma) : \nabla z \,\mathrm{d}x.$$

The same arguments that led to (3.6), now lead to

$$|||\mathrm{Res}|||^2_{Z^\star} \leqslant \sup_{z\in Z} \sum_{T\in\mathcal{T}} \left(\frac{\int_\Omega \mathrm{dev}(q - \mathrm{D}u_h - \mathrm{Curl}\,\gamma) \cdot \nabla z \,\mathrm{d}x}{\|\nabla z\|_{L^2(T)}} + \frac{\int_T (f + \nabla p_h + \mathrm{div}\, q) z \,\mathrm{d}x}{\|\nabla z\|_{L^2(T)}} \right)^2 .$$

Let $u_B \in \mathcal{B}(\mathcal{T};\mathbb{R}^n)$ denote the bubble component of the solution $u_M = u_B + u_h$ and consider the gradient of the piecewise affine part $u_h \in V(\mathcal{T};\mathbb{R}^n)$, i.e. $Du_h \coloneqq \sigma_h - Du_b$. Furthermore, let $u_{h,j} \in P_1(\mathcal{T}) \cap C(\Omega)$ denote the j-th component of u_h and, analogously, $f_j \in L^2(\Omega)$ defines the j-th component of $f \in L^2(\Omega;\mathbb{R}^n)$ for $j = 1, \ldots, n$. The functions

$\varphi_{z,j} \coloneqq e_j \varphi_z$ for the j-th unit vector in $\mathbb{R}^n$,$j = 1, \dots, n$, and the nodal basis function φ_z for all $z \in \mathcal{N}$ define a basis of $\mathcal{P}_1(\mathcal{T}; \mathbb{R}^n) \cap V \subset \mathrm{Mini}(\mathcal{T})$.

Observe that the Galerkin orthogonality of the residual Res from (5.10) and the orthogonality of ∇b_T from (5.9) lead to

$$\begin{aligned} 0 = \mathrm{Res}(\varphi_{z,j}) &= \int_\Omega f_j \varphi_z \,\mathrm{d}x - \int_\Omega \sigma_{h,j} \cdot \nabla \varphi_z \,\mathrm{d}x \\ &= \int_\Omega (f_j + \partial p_h / \partial x_j) \varphi_z \,\mathrm{d}x - \int_\Omega \nabla u_{h,j} \cdot \nabla \varphi_z \,\mathrm{d}x \quad \text{for any } z \in \mathcal{M} \text{ and } j = 1, \dots, n. \end{aligned}$$

This defines the residuals

$$\mathrm{Res}_j(v) \coloneqq \int_\Omega (f_j + \partial p_h / \partial x_j) v \,\mathrm{d}x - \int_\Omega \nabla u_{h,j} \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega) \text{ and } j = 1, \dots, n.$$

Hence, a componentwise application of the equilibration error estimators of Chapter 3 for each component $j = 1, \dots, n$ is possible and leads to some equilibrated quantity $q_{\mathrm{xyz}} \in H(\mathrm{div}, \Omega; \mathbb{R}^{n \times n})$ where $\mathrm{div}\, q_{\mathrm{xyz}} + \widehat{f}$ for $\widehat{f} \coloneqq f + \nabla p_h$ is of higher order. This defines guaranteed upper bounds of $|||\mathrm{Res}|||_{Z^\star}$, e.g. the Braess and Luce-Wohlmuth equilibration error estimators

$$\eta_{\mathrm{B}}^2 \coloneqq \sum_{T \in \mathcal{T}} \left(\|\mathrm{dev}(q_{\mathrm{B}} - \sigma_h)\|_{L^2(T)} + C_{\mathrm{P}}(T) \left\| h_T (\widehat{f} - \widehat{f}_{\mathcal{T}}) \right\|_{L^2(T)} \right)^2,$$

$$\eta_{\mathrm{LW}}^2 \coloneqq \sum_{T \in \mathcal{T}} \left(\|\mathrm{dev}(q_{\mathrm{LW}} - \sigma_h)\|_{L^2(T)} + C_{\mathrm{P}}(T) \left\| h_T (\widehat{f} - \widehat{f}^\star) \right\|_{L^2(T)} \right)^2.$$

The mini fem versions of the error estimators η_{M} and η_{LS} are defined analogously. The postprocessing after Section 3.3 is also applicable componentwise.

5.4 Numerical Experiments for the Mini FEM

This section discusses some numerical benchmark examples. The adaptive mesh refinement in all examples is driven by the Dörfler marking of Subsubsection 2.3.4.2 with the elementwise refinement indicators

$$\begin{aligned} \eta(T)^2 \coloneqq |T| \, \|f + \mathrm{div}\, \sigma_h\|^2_{L^2(T)} + |T|^{1/2} \sum_{E \in \mathcal{E}(T)} \|[\sigma_h \cdot \nu_E]_E\|^2_{L^2(E)} + \|\mathrm{div}\, u_h\|^2_{L^2(T)} / c_0^2 \\ + 0.248 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|^2_{L^2(\partial T \cap \partial\Omega)} / c_0^2. \quad (5.13) \end{aligned}$$

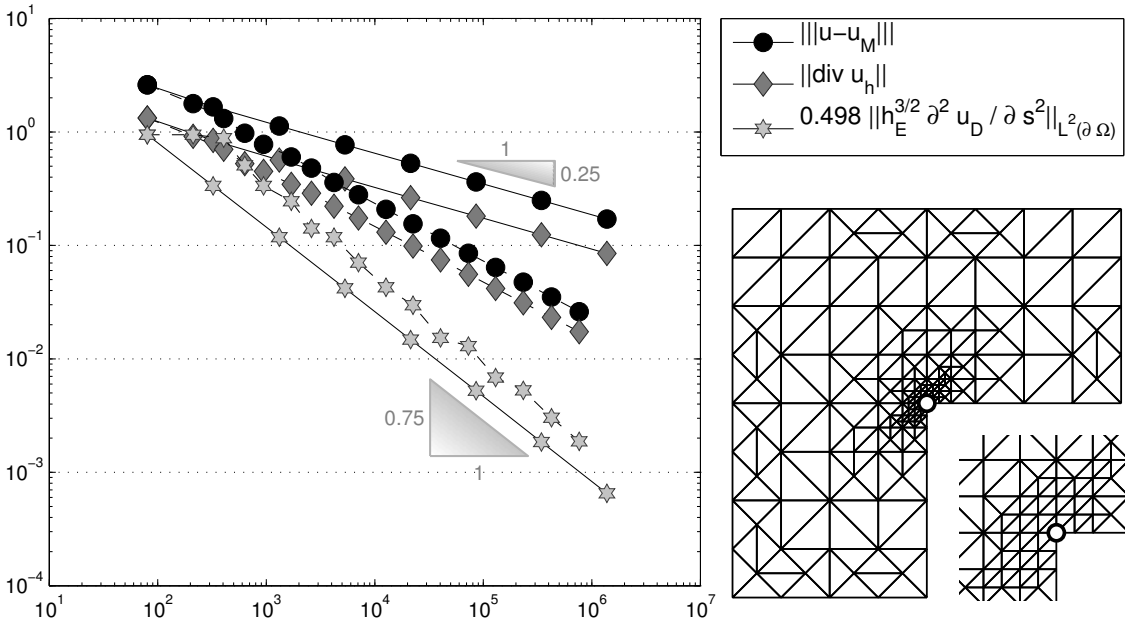

Figure 5.1: Convergence history for the energy error $|||u - u_M|||$, $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.4.1 with respect to the number of degrees of freedom $3|\mathcal{M}| + 2|\mathcal{T}|$. The right image shows the adaptive mesh on level 4 and the neighbourhood of the singular point $(0,0)$ magnified by a factor 2.

5.4.1 L-Shaped Domain

The first benchmark problem employs $f(x,y) \equiv 0$ and the Dirichlet data u_D that matches the exact solution

$$u(r,\varphi) = r^{\alpha} \begin{pmatrix} (\alpha+1)\sin(\varphi)\psi(\varphi) + \cos(\varphi)\psi'(\varphi) \\ -(\alpha+1)\cos(\varphi)\psi(\varphi) + \sin(\varphi)\psi'(\varphi) \end{pmatrix}^T$$

on the L-shaped domain $\Omega = (-1,1)^2 \setminus ([0,1] \times [-1,0])$ with

$$\psi(\varphi) = 1/(\alpha+1)\ \sin((\alpha+1)\varphi)\cos(\alpha\omega) - \cos((\alpha+1)\varphi) \\ + 1/(\alpha-1)\ \sin((\alpha-1)\varphi)\cos(\alpha\omega) + \cos((\alpha-1)\varphi)$$

for $\alpha = 856399/1572864 \approx 0.54$, $\omega = 3\pi/2$ from Verfürth (1989). For the error estimators we set $c_0 = 0.3$ from Stoyan (1999). The streamlines of the velocity field u_h on a finer triangulation is shown in Figure 5.4. The stream flows around the reentrant corner at the point $(0,0)$. This singularity causes the slow experimental convergence rate of about $1/4$ with respect to the number of degrees of freedom $3|\mathcal{M}| + 2|\mathcal{T}|$ for uniform mesh refinement depicted in Figure 5.1.

Figure 5.1 also shows that the adaptive mesh refinement leads to the optimal convergence rate of 0.5 for the energy error $|||u - u_M|||$. The L^2 norm $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ of the divergence of the discrete solution shows the same convergence rate. The threshold

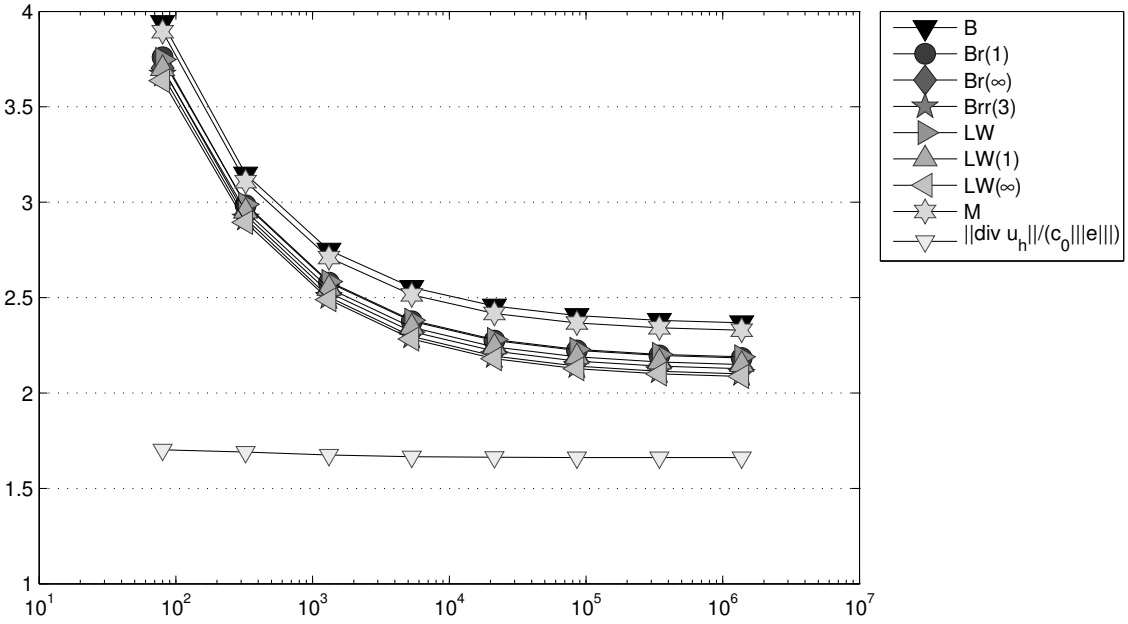

Figure 5.2: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.4.1.

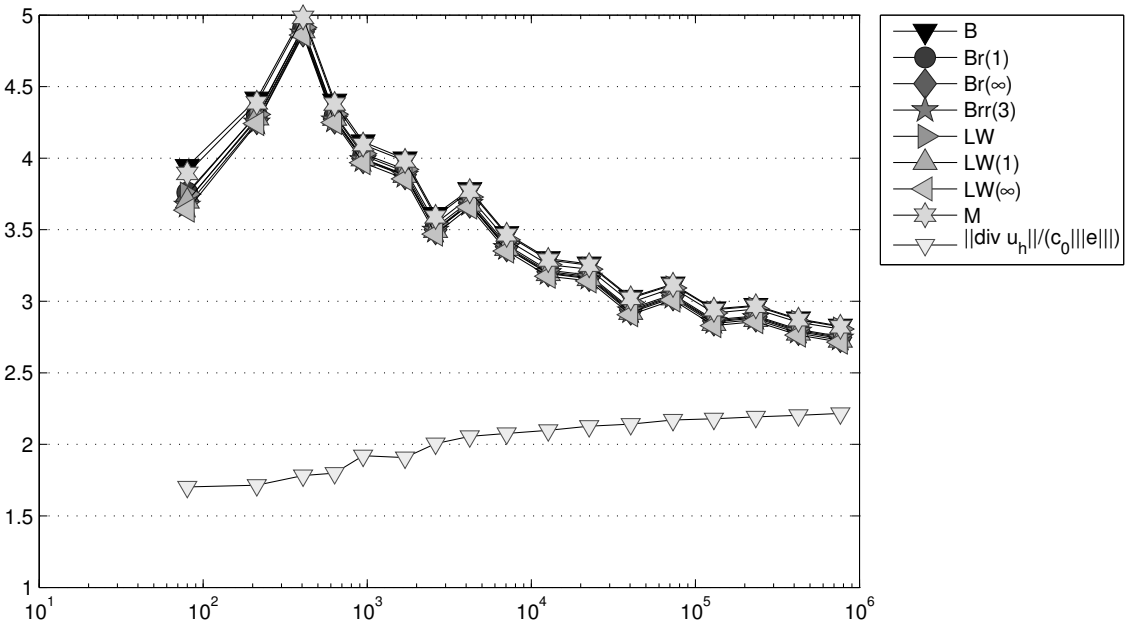

Figure 5.3: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.4.1.

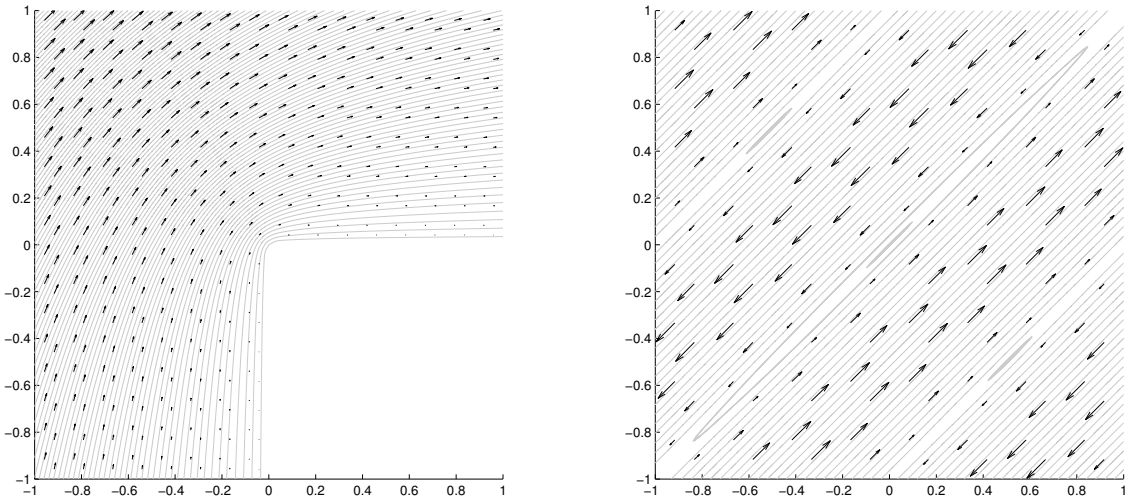

Figure 5.4: Streamlines of the velocity field u_h from the mini FEM for the examples in Subsections 5.4.1 (left) and 5.4.2 (right).

caused by c_0 for the efficiency indices (5.12) is shown in Figures 5.2 and 5.3 and is about 2.2 for adaptive mesh refinement and about 1.7 for uniform mesh refinement. Hence, efficiency indices close to 1 as for the Poisson problems are impossible here. The efficiency indices of η_B are around 2.5 for uniform mesh refinement and about 3 for adaptive mesh refinement. The other estimators perform slightly better. However, there is very little improvement of less than 10 percent by the postprocessing. This appears reasonable, because the contribution that is improved by the postprocessing is only a very small part of the total upper bound. The div u_h contribution dominates with about 70 percent of the total upper bound and cannot be improved by the postprocessing.

5.4.2 Smooth Example on Square Domain

The second benchmark problem employs the right-hand side $f(x,y) = (4\pi^2 \sin(\pi(x - y)), 0)$ and inhomogeneous Dirichlet boundary data u_D with exact solution

$$u(x,y)_j = \sin(\pi x)\cos(\pi y) - \cos(\pi x)\sin(\pi y) \quad \text{for } j = 1,2$$

on the square domain $\Omega = (-1,1)^2$ with $c_0 = 0.3826$ from Stoyan (1999). Figure 5.4 displays the streamlines of the velocity field u_h that are parallel to the main diagonal. There are four streams with alternating flow direction.

Figure 5.5 conveys that the experimental convergence rate of the energy error is optimal for uniform and adaptive mesh refinement. However, the exact energy error on the adaptively generated meshes is a slightly worse. This might be caused by the strong influence of $\|\text{div}\, u_h\|_{L^2(\Omega)}$ to the refinement indicators.

For uniform mesh refinement, the threshold (5.12) caused by the div u_h contribution is significantly lower, but still almost 1. Consequently, the share of $|||\text{Res}|||_\star$ increases and its error estimators lead to better efficiency indices in Figure 5.6. The efficiency indices vary from 2.4 for η_B to 1.8 for η_{LW1}. For adaptive mesh refinement the dominance of the div u_h contribution is higher. As in the first example, the efficiency indices become slightly worse

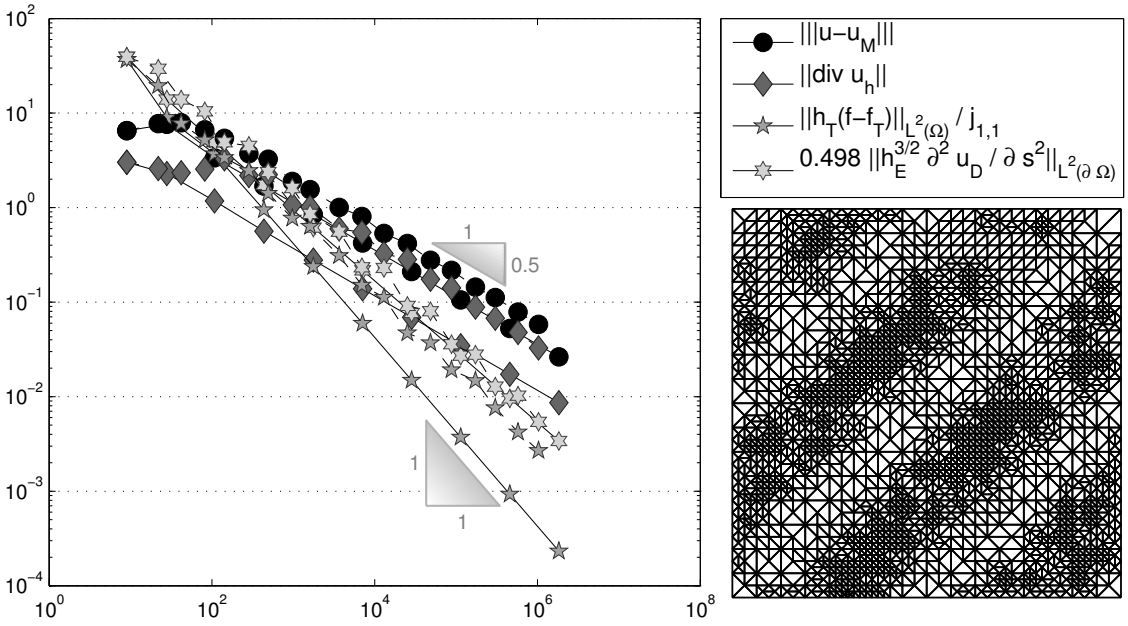

Figure 5.5: Convergence history for the energy error $|||u - u_M|||$, $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.4.2 with respect to the number of degrees of freedom $3|\mathcal{M}| + 2|\mathcal{T}|$. The right image shows the adaptive mesh on level 11.

and the postprocessing less effective (see Figure 5.7). On coarse meshes, the efficiency indices are very large due to the term that measures the error of the inhomogeneous Dirichlet boundary data. The inf-sup constant c_0 in front of this term leads to dramatic overestimation, but the effect reduces on finer meshes thanks to the higher-order property.

5.4.3 Another Smooth Example on Square Domain

The second benchmark problem from Ainsworth and Dörfler (2005) employs the right-hand side $f(x, y) = (-4y, 4x)$ and inhomogeneous Dirichlet boundary data u_D that match the exact solution

$$u(x, y) = (x(1 - x)(1 - 2y), -y(1 - y)(1 - 2x))$$

on the square domain $\Omega = (0, 1)^2$ with $c_0 = 0.3826$ from Stoyan (1999).

In this example the adaptive mesh refinement leads to slightly better meshes as indicated by Figure 5.8. Again, the div u_h contribution forms the main part of the guaranteed upper bound and leads to a threshold (5.12) of around 2.3 for uniform mesh refinement in Figure 5.9 and 1.8 for adaptive mesh refinement in Figure 5.10. The overall efficiency indices decrease from over 5 on coarse meshes to below 3 for the very fine meshes. Unfortunately, the postprocessing is almost ineffective.

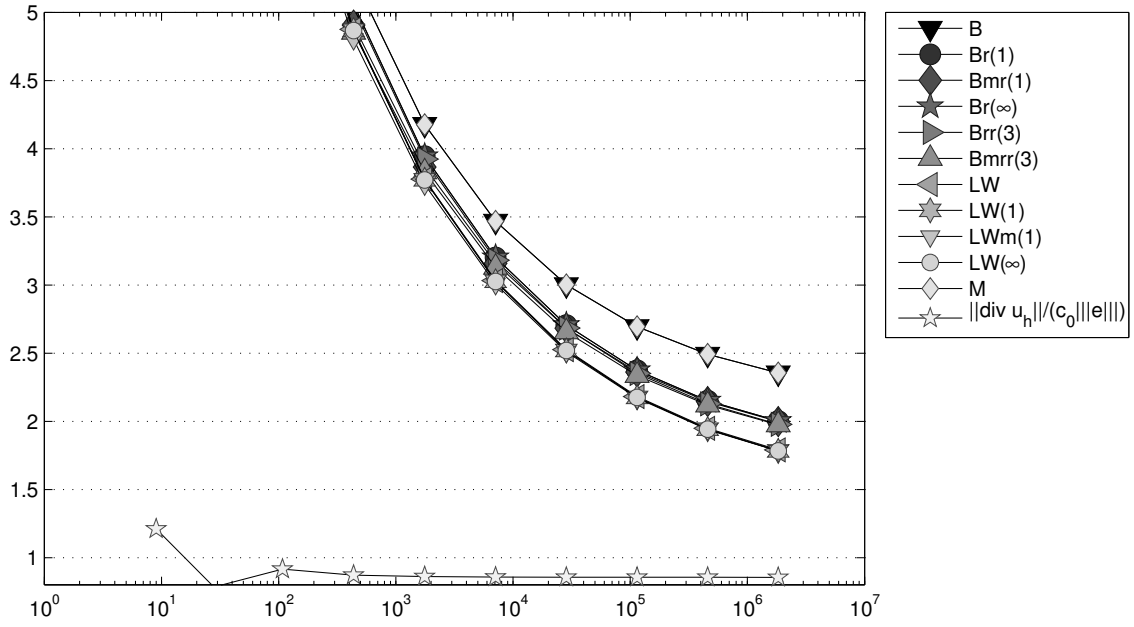

Figure 5.6: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.4.2.

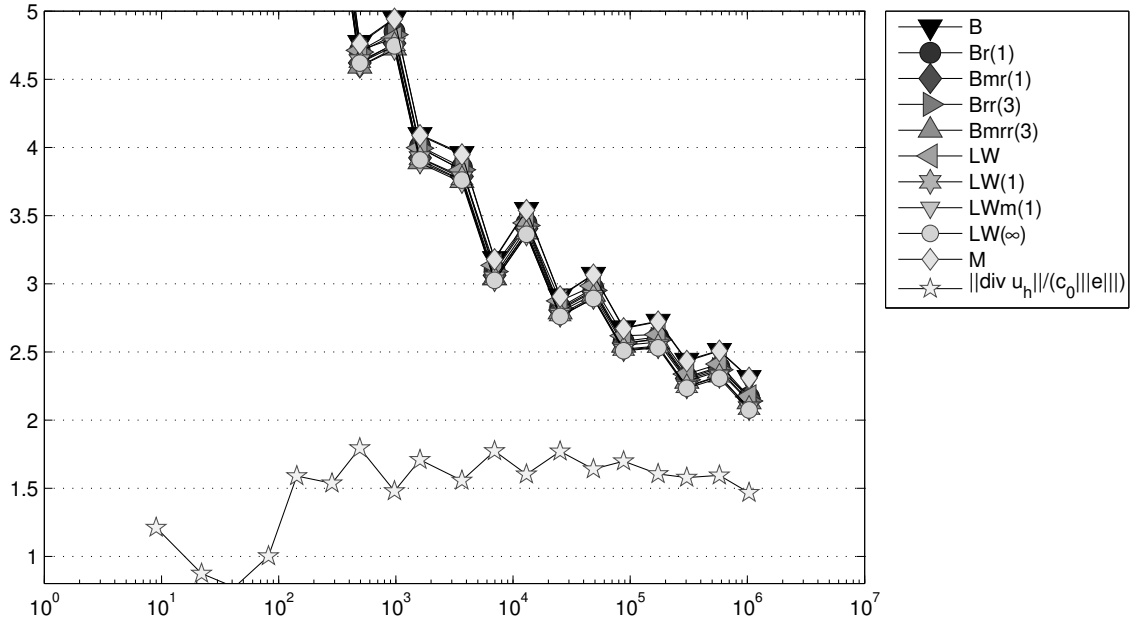

Figure 5.7: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.4.2.

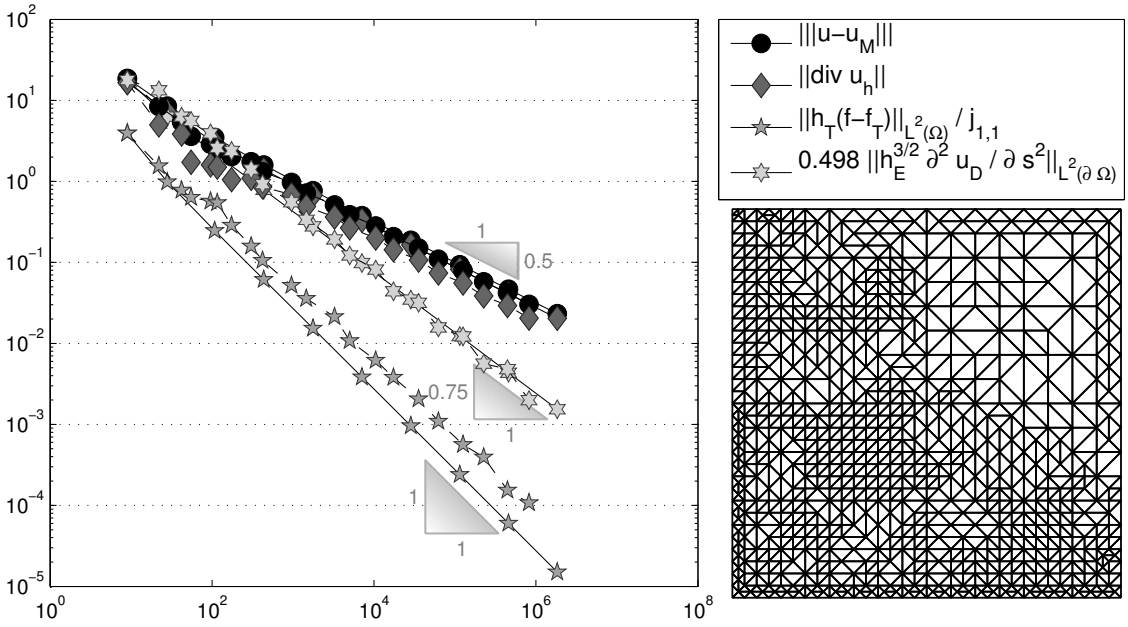

Figure 5.8: Convergence history for the energy error $|||u - u_M|||$, $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.4.3 with respect to the number of degrees of freedom $3|\mathcal{M}| + 2|\mathcal{T}|$. The right image shows the adaptive mesh on level 12.

5.4.4 Colliding Flow

The third benchmark problem employs $f(x,y) \equiv 0$ and the exact solution $u(x,y) = (20xy^4 - 4x^5, 20x^4y - 4y^5)$ on the square domain $\Omega = (-1,1)^2$ with $c_0 = 0.3826$ from Stoyan (1999). This example mimics a colliding flow as depicted in Figure 5.11.

The efficiency indices from Figures 5.13 and 5.14 support all conclusions from the previous experiments.

5.4.5 Backward Facing Step

The last example employs the backward facing step domain $\Omega = ((-2,8) \times (-1,1)) \setminus ([-2,0] \times [-1,0])$, the right-hand side $f \equiv 0$ and the inhomogeneous boundary data

$$u_D(x,y) = \begin{cases} (-y(y-1)/10, 0) & \text{at } x = -2, \\ (-(y^2-1)/80, 0) & \text{at } x = 8. \end{cases}$$

There is no known reference solution, but the example is well-understood (Bank and Welfert, 1991; Carstensen and Funken, 2001).

Figure 5.16 shows the streamlines of the discrete solutions and an adaptive mesh. The refinement detects the singularity at the reentrant corner and also the part of the domain where the velocity of the fluid and the pressure is higher. The right image in Figure 5.16 also shows a characteristic Moffat eddy that is resolved by the finite element method solution.

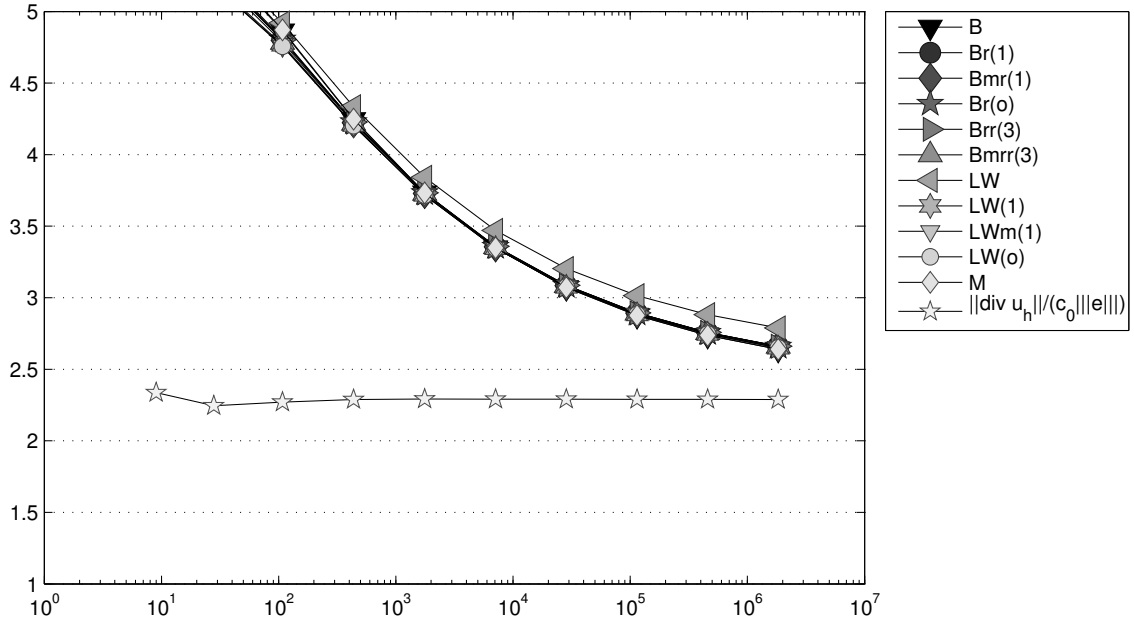

Figure 5.9: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u - u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.4.3.

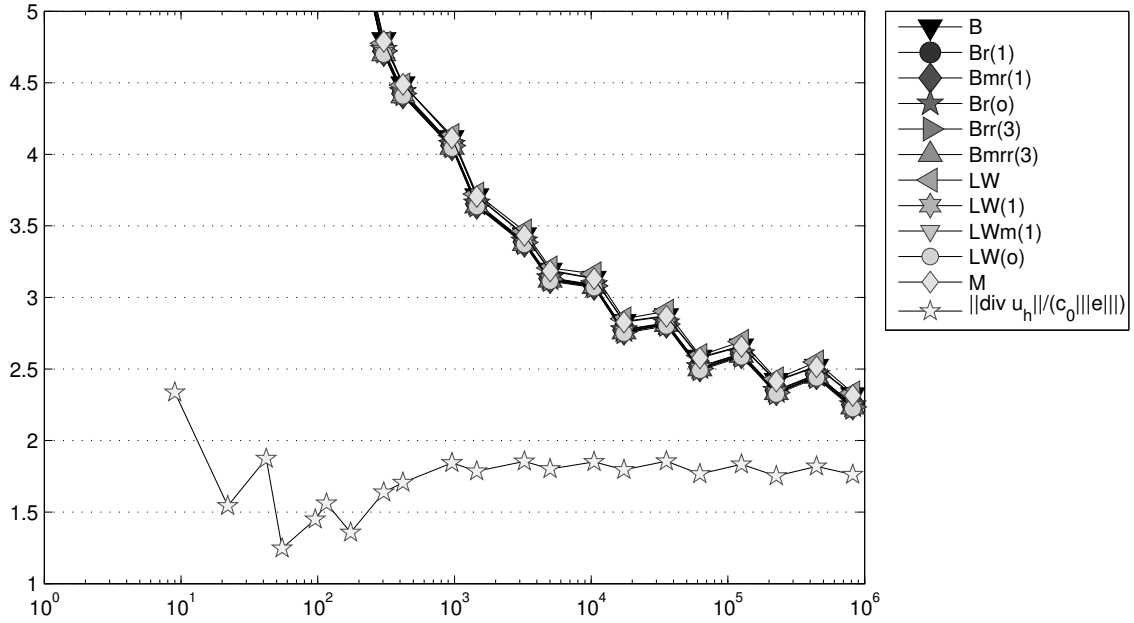

Figure 5.10: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u - u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.4.3.

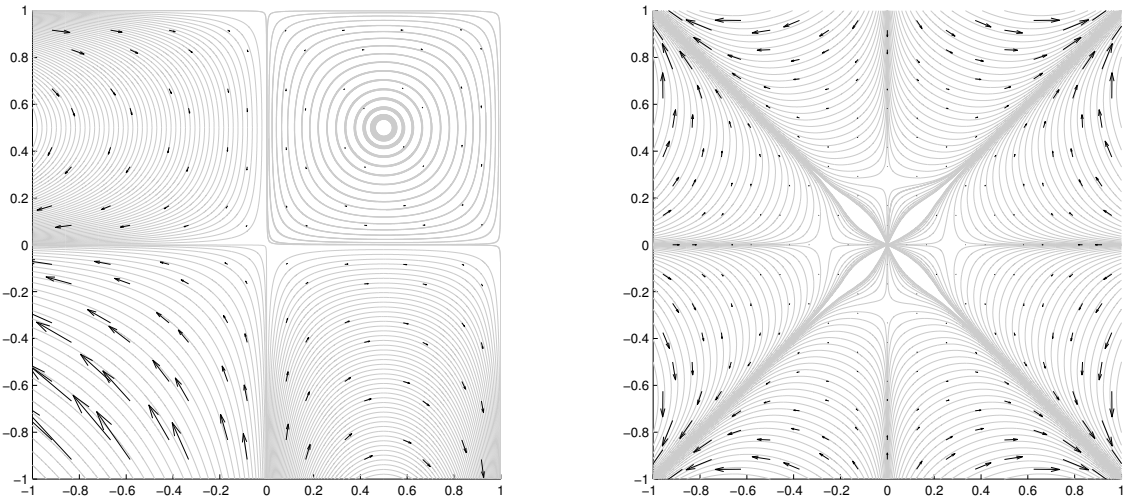

Figure 5.11: Streamlines of the velocity field u_h from the mini FEM for the examples in Subsections 5.4.3 (left) and 5.4.4 (right).

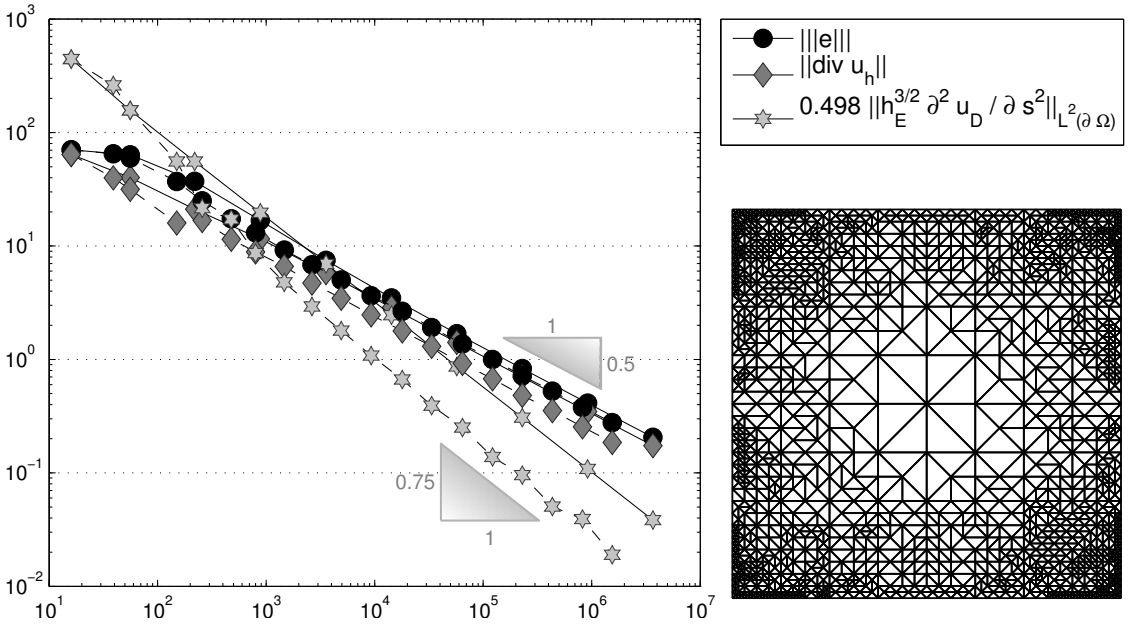

Figure 5.12: Convergence history for the energy error $|||u - u_M|||$, $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.4.4 with respect to the number of degrees of freedom $3|\mathcal{M}| + 2|\mathcal{T}|$. The right image shows the adaptive mesh on level 10.

Moreover, the adaptive mesh refinement leads to an optimal empirical convergence rate of all error estimators (and so implies an optimal convergence rate for the smaller exact energy error) as shown in Figure 5.15.

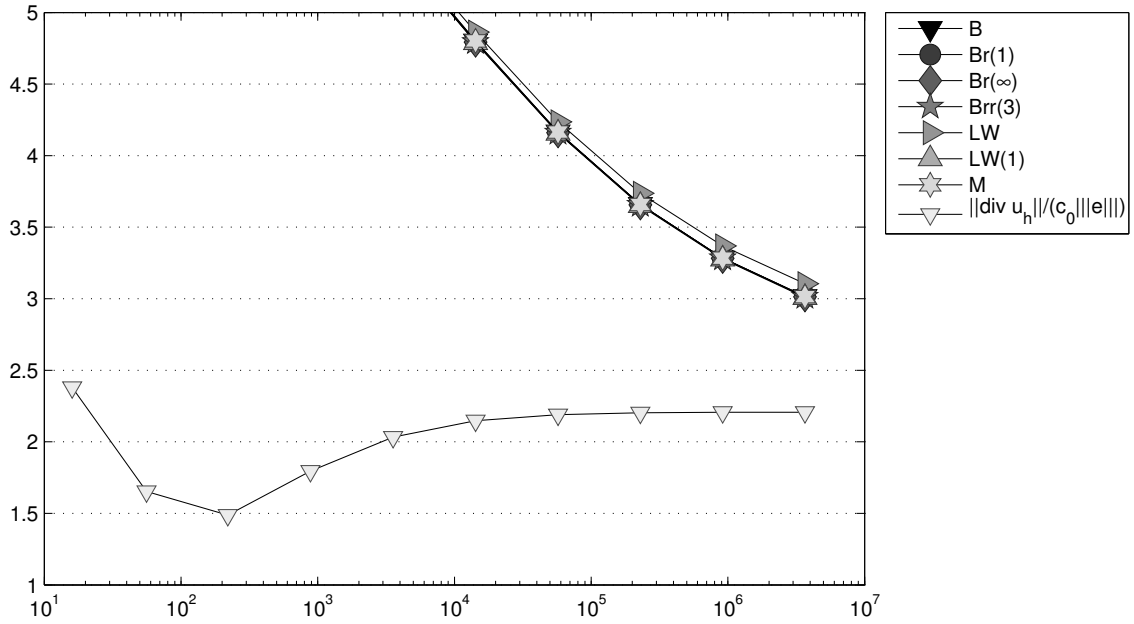

Figure 5.13: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u - u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.4.4.

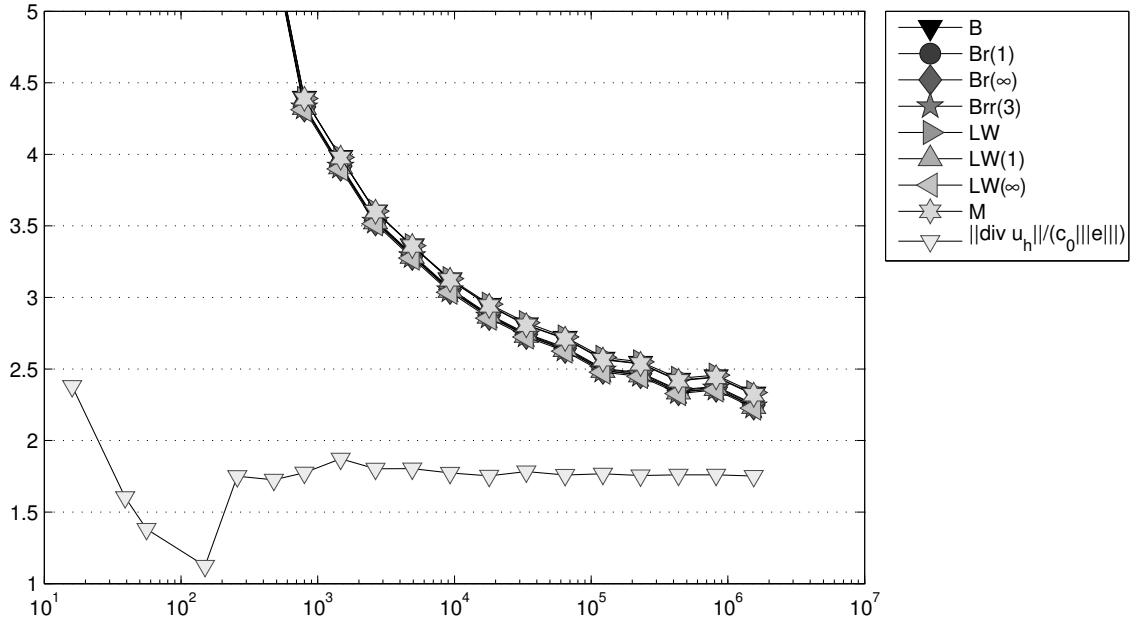

Figure 5.14: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u - u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.4.4.

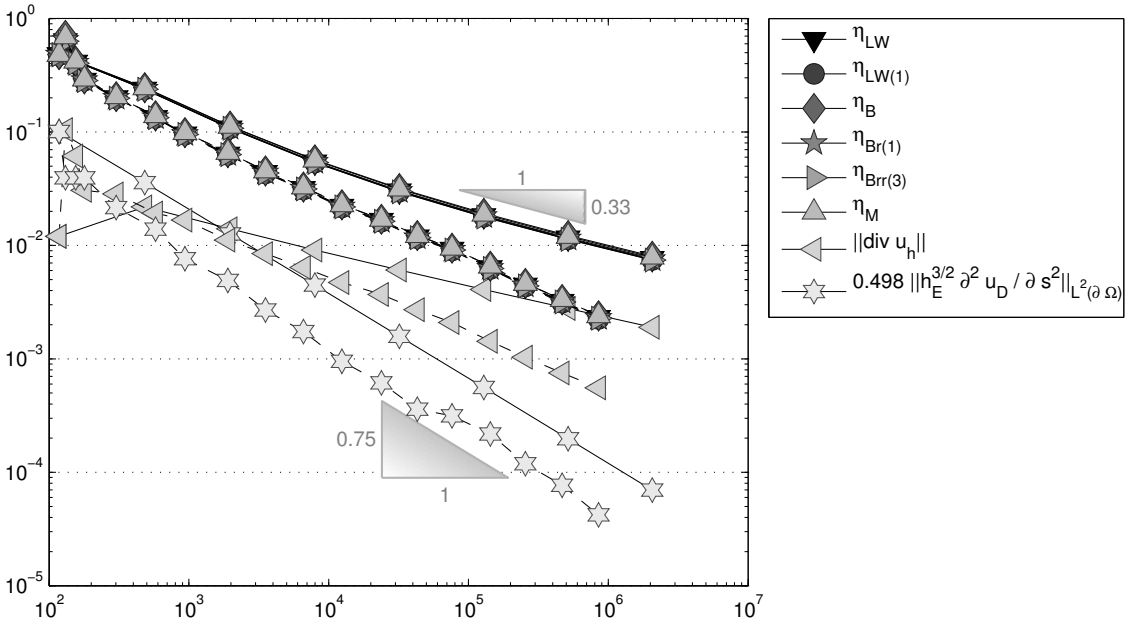

Figure 5.15: Convergence history for the energy error estimator $\eta_{\mathrm{LW}(1)}$, $\|\mathrm{div}\, u_h\|_{L^2(\Omega)}$ and the Dirichlet error contribution $0.4980 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.4.5 with respect to the number of degrees of freedom $3\,|\mathcal{M}| + 2\,|\mathcal{T}|$.

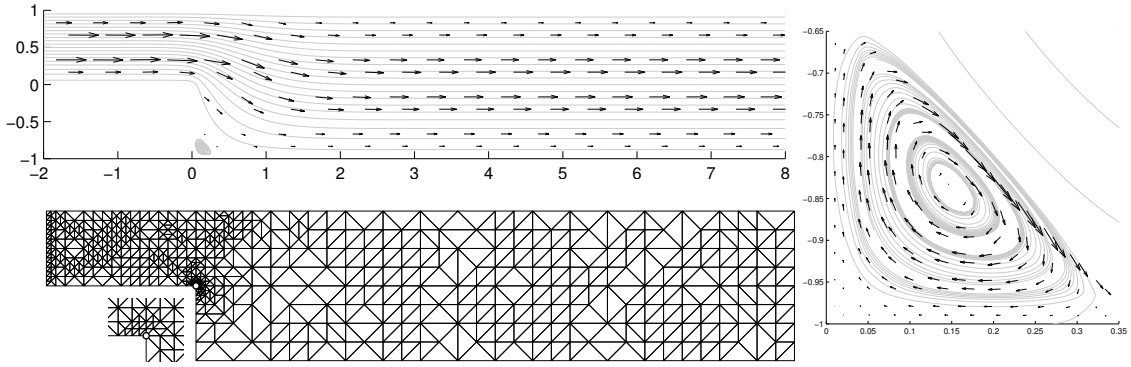

Figure 5.16: The upper left image shows streamlines of the velocity field u_h from the mini FEM for the example in Subsection 5.4.5. The right image shows a zoom-in of the Moffat eddy near the point $(0.15, -0.85)$. The lower left image shows the adaptive mesh on level 8 and the neighbourhood of the singular point $(0, 0)$ magnified by a factor 4.

5.5 A Posteriori Error Control for the Nonconforming CR-FEM

This section aims at guaranteed error control of the nonconforming Crouzeix-Raviart finite element method for the Stokes problem.

5.5.1 Crouzeix-Raviart FEM for the Stokes Equations

The nonconforming formulation of the Stokes problem on some regular triangulation $\mathcal{T}$ involves the broken bilinear forms

$$a_{\mathrm{NC}}(u,v) \coloneqq \int_\Omega \mathrm{D}_{\mathrm{NC}} u : \mathrm{D}_{\mathrm{NC}} v \,\mathrm{d}x \coloneqq \sum_{T\in\mathcal{T}} \int_T \mathrm{D}u : \mathrm{D}v \,\mathrm{d}x \quad \text{for } u,v \in H^1(\mathcal{T};\mathbb{R}^n),$$
$$b_{\mathrm{NC}}(v,q) \coloneqq \int_\Omega q \,\mathrm{div}_{\mathrm{NC}}\, v \,\mathrm{d}x \quad \text{for } q \in L^2_0(\Omega), v \in H^1(\mathcal{T};\mathbb{R}^n)$$

with the piecewise differential operators D_{NC} and $\mathrm{div}_{\mathrm{NC}}$. The discrete counterpart of Z contains divergence-free Crouzeix-Raviart functions and reads

$$Z_{\mathrm{NC}} \coloneqq \{v_{\mathrm{CR}} \in \mathrm{CR}^1_0(\mathcal{T};\mathbb{R}^n) \,\big|\, \mathrm{div}_{\mathrm{NC}}\, v_{\mathrm{CR}} = 0\}.$$

Similar to (5.4), this leads to the pressure-free formulation: seek $u_{\mathrm{CR}} \in u_{D,\mathrm{CR}} + Z_{\mathrm{NC}}$ with

$$u_{D,\mathrm{CR}} = \sum_{E\in\mathcal{E}(\partial\Omega)} \left(\fint_E u_D \,\mathrm{d}s \right) \psi_E$$

and

$$a_{\mathrm{NC}}(u_{\mathrm{CR}}, v_{\mathrm{CR}}) = F(v_{\mathrm{CR}}) \coloneqq \int_\Omega f \cdot v_{\mathrm{CR}} \,\mathrm{d}x \quad \text{for all } v_{\mathrm{CR}} \in Z_{\mathrm{NC}}.$$

Hence, up to boundary conditions, u_{CR} is computed from the Riesz representation of a linear functional in the Hilbert space $(Z_{\mathrm{NC}}, a_{\mathrm{NC}})$. However, the actual implementation employs unconstrained Crouzeix-Raviart elements $v_{\mathrm{CR}} \in \mathrm{CR}^1(\mathcal{T};\mathbb{R}^n)$ and a Lagrange multiplier $p_0 \in \mathcal{P}_0(\mathcal{T})$ to enforce the global constraint

$$\mathrm{div}_{\mathrm{NC}}\, u_{\mathrm{CR}} = 0 \quad \text{a.e. in } \Omega.$$

The discrete problem with Lagrange multipliers seeks $u_{\mathrm{CR}} \in u_{D,\mathrm{CR}} + \mathrm{CR}^1_0(\mathcal{T};\mathbb{R}^n)$, $p_0 \in \mathcal{P}_0(\mathcal{T})$ and $\lambda \in \mathbb{R}$ with

$$\begin{aligned} a_{\mathrm{NC}}(u_{\mathrm{CR}}, v_{\mathrm{CR}}) + b_{\mathrm{NC}}(v_{\mathrm{CR}}, p_0) &= F(v_{\mathrm{CR}}), \\ b_{\mathrm{NC}}(u_{\mathrm{CR}}, q_0) + \lambda \int_\Omega q_0 \,\mathrm{d}x &= 0, \\ \int_\Omega p_0 \,\mathrm{d}x &= 0 \quad \text{for all } v_{\mathrm{CR}} \in \mathrm{CR}^1_0(\mathcal{T};\mathbb{R}^n) \text{ and } q_0 \in \mathcal{P}_0(\mathcal{T}). \end{aligned} \tag{5.14}$$

It turns out that p_h acts also as a discrete pressure and so enters the discrete stress tensor $\sigma_{\mathrm{CR}} \coloneqq \mathrm{D}_{\mathrm{NC}} u_{\mathrm{CR}} + p_h \mathbb{I}$. Listing 5.2 displays the `MATLAB` code that computes and solves the linear system of equations in (5.14). The overall structure of the code is the same as for the mini finite element method from Subsection 5.2.1 and Listing 5.1. The sole difference is the set of basis functions with consequences for the enumeration of the degrees of freedom in Lines 19,25 and 26 and the different gradients in Line 14.

```
function [u,p,A,b,nrDof]=solveCRP0Stokes(f,u4Db,c4n,n4e,n4sDb,degree_f,degree_u4Db)
n4s = computeN4s(n4e);
s4n = computeS4n(n4e,n4s);
s4e = computeS4e(n4e);
Dbs = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
l4DbS = computeLength4s(c4n,n4sDb);
nrElems = size(n4e,1);
nrSides = size(n4s,1);
A4e = zeros(3,3,nrElems);
B4e = zeros(1,6,nrElems);
grads4e = zeros(3,2,nrElems);
area4e = computeArea4e(c4n,n4e);
for j=1:nrElems
    grads = -2*([ones(1,3);c4n(n4e(j,:),:)']\[zeros(1,2);eye(2)]);
    A4e(:,:,j) = area4e(j)*(grads*grads');
    B4e(:,:,j) = area4e(j)*grads(:)';
    grads4e(:,:,j) = grads;
end
dofs_u = s4e(:,[2 3 1])';
I = repmat(dofs_u(:),1,size(dofs_u,1))';
J = repmat(dofs_u',1,size(dofs_u,1))';
A = sparse(I(:),J(:),A4e(:));
A = [A sparse(nrSides,nrSides)
     sparse(nrSides,nrSides) A];
dofs_u = [s4e(:,[2 3 1]) nrSides+s4e(:,[2 3 1])];
dofs_p = [1:nrElems];
I = repmat(dofs_p(:),1,size(dofs_u,2))';
J = repmat(dofs_u,1,size(dofs_p,1))';
B = sparse(I(:),J(:),B4e(:),nrElems,2*nrSides);
integrand = @(n4p,pts,pts_ref)RHS_BasisXf(n4p,pts,pts_ref,f);
b4e = integrate(c4n,n4e,integrand,degree_f+1);
b = accumarray(dofs_u(:),b4e(:));
b = [b; zeros(nrElems+1,1)];
freeSides=setdiff(1:size(n4s,1),Dbs);
mean4Dbs = integrate(c4n,n4sDb,@(x,y,z)(u4Db(y)),degree_u4Db)./[l4DbS l4DbS];
x = zeros(2*nrSides+nrElems+1,1);
x([Dbs nrSides+Dbs])=[mean4Dbs(:,1)' mean4Dbs(:,2)'];
b = b - [A B' sparse(2*nrSides,1)]'*x(1:2*nrSides);
dofs =[freeSides nrSides+freeSides 2*nrSides + (1:nrElems) 2*nrSides+nrElems+1];
M=[A B' sparse(2*nrSides,1);B sparse(nrElems,nrElems) area4e;...
    sparse(1,2*nrSides) area4e' 0];
x(dofs)=M(dofs,dofs)\b(dofs);
nrDof = length(dofs);
u1 = x(1:nrSides);
u2 = x(nrSides+1:2*nrSides);
u=[u1 u2];
p=x(2*nrSides+1:2*nrSides+nrElems);
end

function val = RHS_BasisXf(n4p,pts,pts_ref,f)
  x = pts_ref(1); y = pts_ref(2);
  f4pts = f(pts);
  f4pts = f4pts(:,[1 1 1 2 2 2]);
  basis4pts = repmat([1-2*y -1+2*(x+y) 1-2*x],[size(pts,1) 2]);
  val = f4pts.*basis4pts;
end
```

Listing 5.2: Listing for the Crouzeix-Raviart nonconforming fem solver `solveCRP0Stokes` for the Stokes problem.

5.5.2 Error Decomposition

The error decomposition is based on Ainsworth and Dörfler (2005), but the results here are slightly sharper and involve a residual of the form (3.1) with $\sigma_h = \sigma_{\mathrm{CR}}$ and n components, i.e.,

$$\mathrm{Res}(v) = \int_\Omega f \cdot v \,\mathrm{d}x - \int_\Omega \sigma_{\mathrm{CR}} : \mathrm{D}v \,\mathrm{d}x \quad \text{for } v \in Z. \tag{5.15}$$

Its dual norm with respect to $Z^\star$ and the usual energy norm $|||\cdot||| := \|\mathrm{D}\cdot\|_{L^2(\Omega)}$ reads

$$|||\mathrm{Res}|||_{Z^\star} := \sup_{v \in Z\setminus\{0\}} \mathrm{Res}(v)/|||v|||\,.$$

The following Theorem allows the design of guaranteed upper bounds for $|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}$ by conforming interpolations as for the Poisson problem in Theorem 4.4.1.(b).

Theorem 5.5.1. *(a) It holds*

$$|||u - u_{\mathrm{CR}}|||^2_{\mathrm{NC}} \leqslant |||\mathrm{Res}|||^2_{Z^\star} + \min_{\substack{v \in H^1(\Omega;\mathbb{R}^n) \\ v = u_D \text{ along } \partial\Omega}} \Big(\|\mathrm{D}_{\mathrm{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)} + 1/c_0 \ \|\mathrm{div}\, v\|_{L^2(\Omega)} \Big)^2$$

with the residual Res *from* (5.15).
(b) The quantity

$$\eta_0^2 := \sum_{T \in \mathcal{T}} \Big(C_{\mathrm{P}}(T) \, \|h_T(f - f_T)\|_{L^2(T)} + \|f_{\mathcal{T}}/n \otimes (\bullet - \mathrm{mid}(T))\|_{L^2(T)} \Big)^2$$

is an upper bound for $|||\mathrm{Res}|||_{Z^\star} \leqslant \eta_0$.

Proof. Proof of (a). The point of departure is the orthogonal split

$$\mathrm{D}_{\mathrm{NC}} e = \mathrm{D}z + y$$

from Lemma 5.1.1.(a) into some $z \in Z$ and $y \in Y$, such that

$$|||u - u_{\mathrm{CR}}|||^2_{\mathrm{NC}} = |||z|||^2_{\mathrm{NC}} + \|y\|^2_{L^2(\Omega)} = \int_\Omega \mathrm{D}_{\mathrm{NC}} e : \mathrm{D}z \,\mathrm{d}x + \int_\Omega \mathrm{D}_{\mathrm{NC}} e : y \,\mathrm{d}x.$$

Since $\mathbb{I} : \mathrm{D}v = \mathrm{div}\, v$ for any $v \in Z$, it holds

$$\begin{aligned} \mathrm{Res}(v) &= \int_\Omega (\sigma - \sigma_{\mathrm{CR}}) : \mathrm{D}v \,\mathrm{d}x = \int_\Omega (\mathrm{D}_{\mathrm{NC}} e + (p - p_h)\mathbb{I}) : \mathrm{D}v \,\mathrm{d}x = \int_\Omega \mathrm{D}_{\mathrm{NC}} e : \mathrm{D}v \,\mathrm{d}x \\ &= \int_\Omega \mathrm{D}z : \mathrm{D}v \,\mathrm{d}x \leqslant |||z|||\,|||v|||\,. \end{aligned}$$

This shows $|||\mathrm{Res}|||_{Z^\star} \leqslant |||z|||$. The equality $|||\mathrm{Res}|||_{Z^\star} = |||z|||$ follows from the particular choice $v = z$.

To estimate $\int_\Omega \mathrm{D}_{\mathrm{NC}} e : y \,\mathrm{d}x$, consider any $v \in H^1(\Omega;\mathbb{R}^n)$ with $u - v = 0$ on $\partial\Omega$. Together

with Lemma 5.1.1.(b), it follows

$$\begin{aligned}\int_\Omega \mathrm{D_{NC}} e : y \,\mathrm{d}x &= \int_\Omega \mathrm{D_{NC}}(u_{\mathrm{CR}} - v) : y \,\mathrm{d}x + \int_\Omega \mathrm{D}(v-u) : y \,\mathrm{d}x \\ &\leqslant \|\mathrm{D_{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)} \, \|y\|_{L^2(\Omega)} + \int_\Omega \mathrm{div}(v-u) w \,\mathrm{d}x \\ &\leqslant \left(\|\mathrm{D_{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)} + 1/c_0 \, \|\mathrm{div}\, v\|_{L^2(\Omega)} \right) \|y\|_{L^2(\Omega)} .\end{aligned}$$

This concludes the proof of (a).

The proof of (b) is very similar to the proof of Theorem 4.4.1.(c) □

Listing 4.1 computes the quantity η_0 for $n = 2$ dimensions and the next section explains modifications to the interpolation designs of Subsection 4.4.2 for the computation of guaranteed upper bounds for the second term on the right-hand side of Theorem 5.5.1.(a).

5.6 Modifications to Interpolation Designs in Presence of Divergence Constraint

Naturally, the componentwise application of the conforming interpolations of Subsection 4.4.2.2 leads to some $v_{\mathrm{xyz}} \in H^1(\Omega;\mathbb{R}^n)$ with $v_{\mathrm{xyz}} = u_D$ along $\partial\Omega$ and, together with Theorem 5.5.1.(a), to the associated guaranteed upper bound

$$|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}^2 \leqslant \eta_0^2 + \left(\left\|\mathrm{D_{NC}}(u_{\mathrm{CR}} - v_{\mathrm{xyz}})\right\|_{L^2(\Omega)} + 1/c_0 \, \left\|\mathrm{div}\, v_{\mathrm{xyz}}\right\|_{L^2(\Omega)} \right)^2 . \tag{5.16}$$

However, the term in the brackets on the right-hand side consists of the sum of two norms and an optimal interpolation in discrete subspace $W_h \subset H^1(\Omega;\mathbb{R}^n)$ has to minimise this sum and not the sum of squares. As in Subsection 3.2.5, a Young inequality shows

$$\begin{aligned}&\min_{\substack{v\in W_h \\ v|_{\partial\Omega} = u_D}} \left(\|\mathrm{D_{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)} + 1/c_0 \, \|\mathrm{div}\, v\|_{L^2(\Omega)} \right)^2 \\ &\qquad = \min_{\substack{v\in W_h \\ v|_{\partial\Omega} = u_D}} \min_{\lambda>0} \left((1+\lambda) \, \|\mathrm{D_{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)}^2 + (1+1/\lambda)/c_0 \, \|\mathrm{div}\, v\|_{L^2(\Omega)}^2 \right) \\ &\qquad =: \min_{\substack{v\in W_h \\ v|_{\partial\Omega} = u_D}} \min_{\lambda>0} \widehat{M}(\lambda, v).\end{aligned}$$

Algorithm 5.1 exploits this relation and performs k iterations (usually $k = 3$). In each iteration the least-square problem

$$v_{\mathrm{xyz}} := \mathrm{argmin} \left\{ \widehat{M}(\lambda, v) \,\middle|\, v \in W_h, v|_{\partial\Omega} = u_D \right\}$$

is solved in W_h and then the optimal λ is calculated (recall that $(a+b)^2 = (1+\lambda)a^2 + (1+1/\lambda)b^2$ for the optimal $\lambda = b/a$). If $W_h = P_2(\mathcal{T};\mathbb{R}^n) \cap C(\Omega)$, the outcome is called η_{MP2}

Set $\lambda \coloneqq 1$;
for $\ell \coloneqq 1, \dots, k$ **do**
  Compute $v_{\mathrm{xyz}} \coloneqq \operatorname{argmin}\left\{\widehat{M}(\lambda, v) \,\middle|\, v \in W_h, v|_{\partial\Omega} = u_D\right\}$;
  Update $\lambda \coloneqq \|\operatorname{div} v\|_{L^2(\Omega)} / (c_0 \|\mathrm{D}_{\mathrm{NC}}(u_{\mathrm{CR}} - v)\|_{L^2(\Omega)})$;
  if *linear system is nearly singular* **then**
    break;
Output: $\eta_{\mathrm{xyz}} \coloneqq \eta_0^2 + \widehat{M}(\lambda, v_{\mathrm{xyz}})$

Algorithm 5.1: Algorithm for the minimisation of the sum of norms in (5.16) for some discrete subspace $W_h \subset V$. Usually the number of iterations k equals three in the numerical examples below.

Set $\lambda \coloneqq 1$ and $v_{\mathrm{PMRED}} = v_{\mathrm{ARED}} \in \mathcal{P}_1(\mathrm{red}(\mathcal{T}); \mathbb{R}^n) \cap C(\Omega, \mathbb{R}^n)$;
for $\ell \coloneqq 1, 2, 3$ **do**
  for $z \in \mathcal{M}$ **do**
    $v_z = \operatorname{argmin}_{\alpha \in \mathbb{R}^n} \widehat{M}(\lambda, v_{\mathrm{PMRED}} + \alpha \varphi_z^{\mathrm{red}})$;
    Modify $v_{\mathrm{PMRED}}(z) = v_{\mathrm{PMRED}}(z) + v_z$;
  Update $\lambda \coloneqq \|\operatorname{div} v_{\mathrm{PMRED}}\|_{L^2(\Omega)} / (c_0 \|\mathrm{D}_{\mathrm{NC}}(u_{\mathrm{CR}} - v_{\mathrm{PMRED}})\|_{L^2(\Omega)})$;
Output: $\eta_{\mathrm{PMRED}} \coloneqq \eta_0^2 + \widehat{M}(\lambda, v_{\mathrm{PMRED}})$

Algorithm 5.2: Algorithm for the computation of the Stokes variant of the error estimator η_{PMRED}.

and $W_h = P_1(\mathrm{red}(\mathcal{T}); \mathbb{R}^n) \cap C(\Omega)$ yields η_{MP1RED}. Of course, instead of solving the least-square problem, a preconditioned conjugated gradients scheme can be applied with initial value v_{ARED}. The result after three cg iterations (three cg iterations in each λ iteration, so altogether nine cg iterations) is called $\eta_{\mathrm{MP2cg(3)}}$ for $W_h = P_2(\mathcal{T}; \mathbb{R}^n) \cap C(\Omega)$ or $\eta_{\mathrm{MP1REDcg(3)}}$ for $W_h = P_1(\mathrm{red}(\mathcal{T}); \mathbb{R}^n) \cap C(\Omega)$. The example in Subsection 5.7.2 below includes a comparison of efficiency indices for different choices of k to support the reasonable choice $k = 3$.

The error estimator η_{PMRED} from Subsection 4.4.2.2 can can also be modified to cope with the sum of the two norms. To do so, the values v_z in (4.19) are improved by an iteration similar to Algorithm 5.2.

5.6.1 Treatment of Inhomogeneous Boundary Data

The designs of the test function v in Section 5.6 may not resolve the inhomogeneous boundary data exactly. To qualify them as a valid upper bound in the sense of Theorem 5.5.1.(a) we apply Theorem 4.2.2 componentwise for $v_D = u_D - v|_{\partial\Omega}$ and design some $w_D \in H^1(\Omega; \mathbb{R}^n)$ with $w_D = v_D$ on $\partial\Omega$. Since w_D satisfies $u - (v + w_D) \in H^1_D(\Omega; \mathbb{R}^n)$, $v + w_D$ is a valid interpolation with correct boundary data and leads to the upper bound

$$|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}^2$$

$$\leqslant \eta_0^2 + \left(\left\| \mathrm{D}_{\mathrm{NC}}(u_{\mathrm{CR}} - v_{\mathrm{xyz}} - w_D) \right\|_{L^2(\Omega)} + 1/c_0 \, \left\| \mathrm{div}(v_{\mathrm{xyz}} + w_D) \right\|_{L^2(\Omega)} \right)^2$$
$$\leqslant \eta_0^2 + \left(\left\| \mathrm{D}_{\mathrm{NC}}(u_{\mathrm{CR}} - v_{\mathrm{xyz}}) \right\|_{L^2(\Omega)} + 1/c_0 \, \left\| \mathrm{div}\, v_{\mathrm{xyz}} \right\|_{L^2(\Omega)} + (1 + 1/c_0) \, |||w_D||| \right)^2 .$$

The energy norm estimate for w_D from Theorem 4.2.2 yields

$$\|\mathrm{div}\, w_D\|_{L^2(\Omega)} \leqslant |||w_D||| \leqslant C_{\mathrm{D},1}(\mathcal{E}(\partial\Omega)) \left\| h_{\mathcal{E}}^{3/2} \partial_{\mathcal{E}}^2 (u_D - v)/\partial s^2 \right\|_{L^2(\partial\Omega)} .$$

For the more elaborate designs v_{ARED}, v_{PMRED}, v_{MP1RED}, $v_{\mathrm{MP1REDcg(3)}}$,v_{MP2} or $v_{\mathrm{MP2cg(3)}}$ based on $\mathcal{P}_1(\mathrm{red}(\mathcal{T}))$ or $\mathcal{P}_2(\mathcal{T})$, the design of w_D in Theorem 4.2.2 can be performed on $\mathrm{red}(\mathcal{T})$ with halved edge lengths, because $v_D \in H_0^1(E)$ for all Dirichlet boundary edges $E \in \mathcal{E}^{\mathrm{red}}(\partial\Omega)$ of $\mathrm{red}(\mathcal{T})$. This leads to the improved constant $C_{\mathrm{D},1}(\mathcal{E}^{\mathrm{red}}(\partial\Omega)) = 0.4980/2^{3/2} = 0.1761$ for triangulations that consist of right isosceles triangles.

5.7 Numerical Experiments for Nonconforming CR-FEM

The adaptive mesh refinement in all examples is driven by the Dörfler marking of Subsubsection 2.3.4.2 with the edge-based refinement indicators

$$\eta(E)^2 := h_E^2 \, \|f\|_{L^2(\omega_E)}^2 + h_E \, \|[\sigma_{\mathrm{CR}} \cdot \tau_E]\|_{L^2(E)}^2 + 0.248(1 + 1/c_0^2) \left\| h_E^{3/2} \partial^2 u_D/\partial s^2 \right\|_{L^2(E \cap \partial\Omega)}^2 . \tag{5.17}$$

5.7.1 L-Shaped Domain

This subsection revisits the L-shaped domain example from Subsection 5.4.1. The nonconforming finite element method for the Stokes problem has one big advantage. Since the ansatz functions are divergence-free, the efficiency of the a posteriori error estimators is not polluted. Figures 5.18 and 5.19 show that the efficiency indices can be close to 1 for more expensive error estimators like η_{MP2}. This suggests that the upper bound for the first residual $|||\mathrm{Res}|||_{Z^\star} \leqslant \eta_0$ is also sharp in this application. Figure 5.18 shows that the averaging η_{ARED} on $\mathrm{red}(\mathcal{T})$ can be worse than η_{A} on $\mathcal{T}$. This is due to the additional div term of the conforming interpolation in Theorem 5.5.1.(a). The improved averaging η_{PMRED} allows highly improved efficiency indices. In Figure 5.19 its efficiency indices drop below 2.7 while η_{A} and η_{ARED} stay around 3.5. However, for good efficiency indices below 2, it seems more beneficial to use $\eta_{\mathrm{MP2cg(3)}}$.

5.7.2 Smooth Example on Square Domain

For the example from Subsection 5.4.2, Figure 5.20 shows the convergence rates of the energy error and the Dirichlet error contribution and a adaptive mesh. The efficiency indices of the error estimators for the nonconforming finite element method are similar to the efficiency indices in the previous example. Again, η_{ARED} is worse than η_{A} and also the improvement η_{PMRED} hardly shows improvements over η_{A} as shown in Figures 5.21

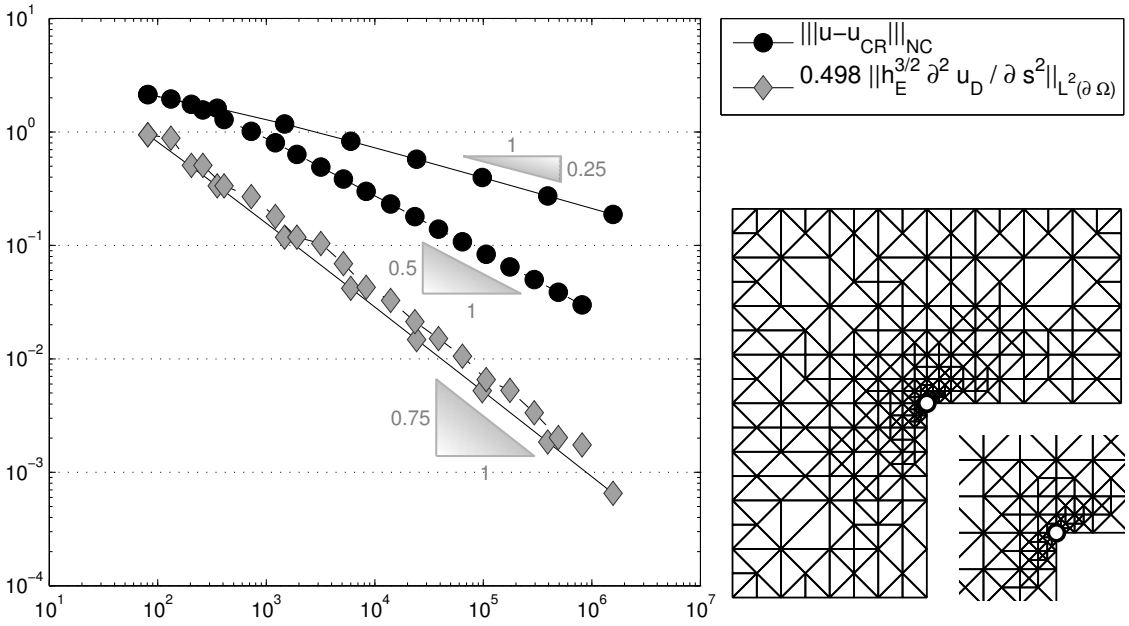

Figure 5.17: Convergence history for the energy error $|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.7.1. The right image shows the adaptive mesh on level 7.

ndof	k=1	k=2	k=3	k=4	k=5	k=10
5	5.085	5.085	5.085	5.085	5.085	5.085
25	2.412	2.372	2.361	2.356	2.354	2.351
113	1.976	1.931	1.918	1.913	1.911	1.907
481	1.594	1.553	1.544	1.541	1.540	1.540
1985	1.430	1.389	1.384	1.383	1.383	1.382
8065	1.356	1.307	1.302	1.302	1.302	1.302
32513	1.323	1.267	1.260	1.260	1.260	1.260
130561	1.308	1.247	1.238	1.237	1.237	1.237

Table 5.1: Efficiency indices for η_{MP2} for different number of iterations k in Algorithm 5.1 for the minimisation of the sum of norms on uniform meshes for the example from Subsection 5.7.2 with respect to the number of degrees of freedom (ndof).

and 5.22. However, in case of uniform mesh refinement their efficiency indices approach 1.5, which is almost as good as the optimal error estimators and their cg approximations. For adaptive mesh refinement the gap between η_{PMRED} and $\eta_{\mathrm{MP2cg(3)}}$ is almost 1 and quite large. So also this experiment supports the observation that the pcg approximations of the optimal quadratic interpolation η_{MP2} are superior. Table 5.1 compares different realisations of η_{MP2} after Algorithm 5.1 with $k \neq 3$ iterations for the minimisation of the sum of norms on uniform refined meshes. The main observation is that there is a significant reduction by $k = 3$ iterations compared to $k = 1$ iterations. However, more iterations do not show much further improvement. Also for adaptive mesh refinement,

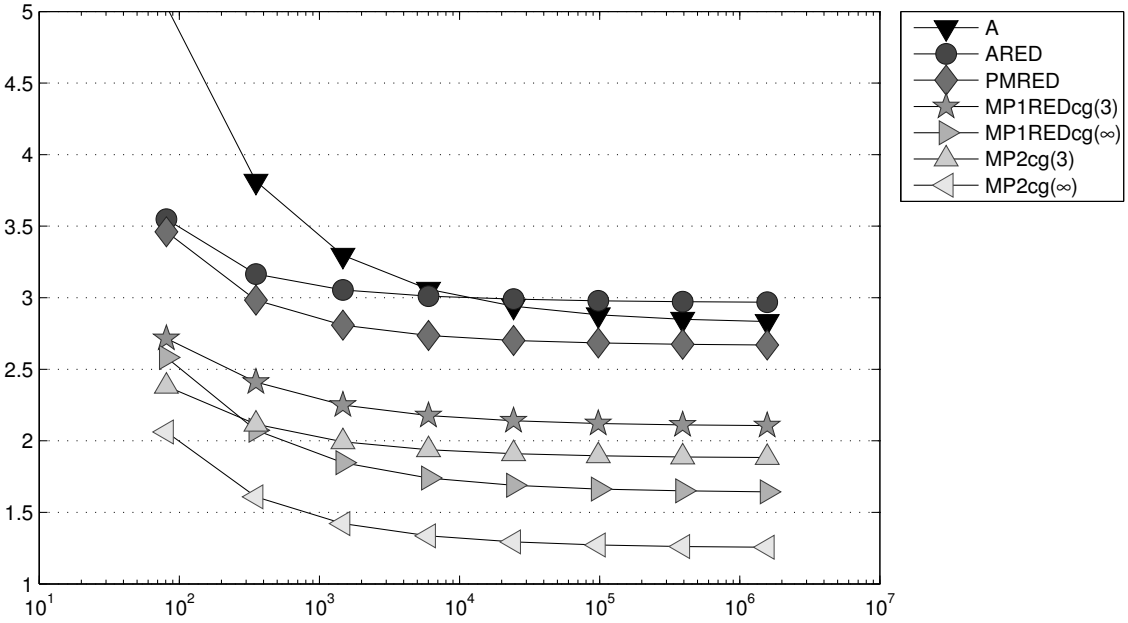

Figure 5.18: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.7.1.

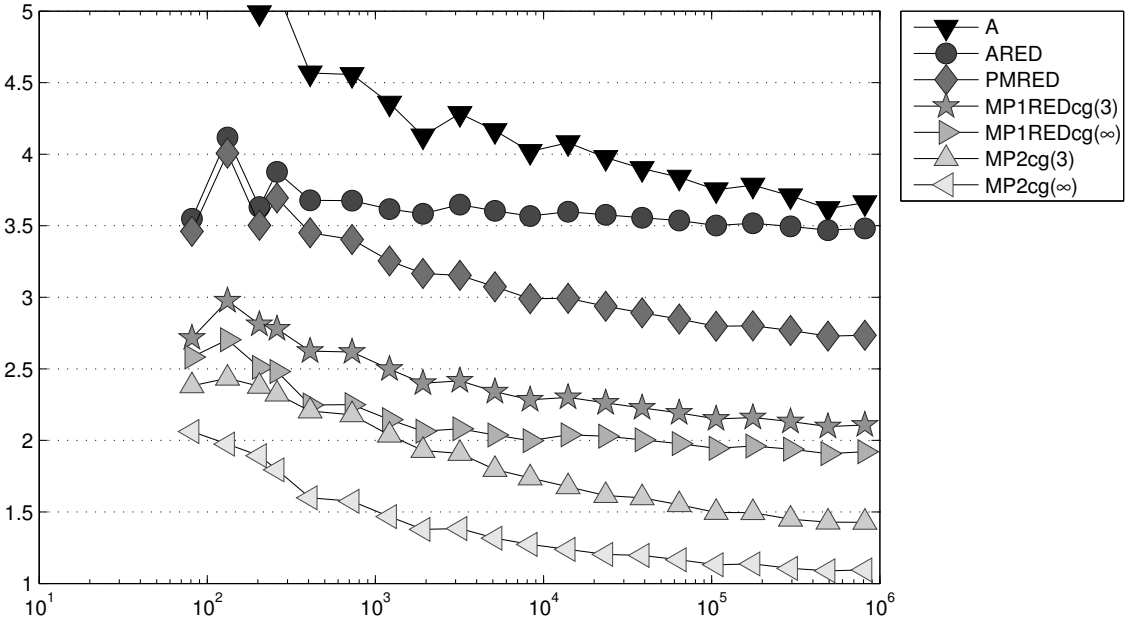

Figure 5.19: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.7.1.

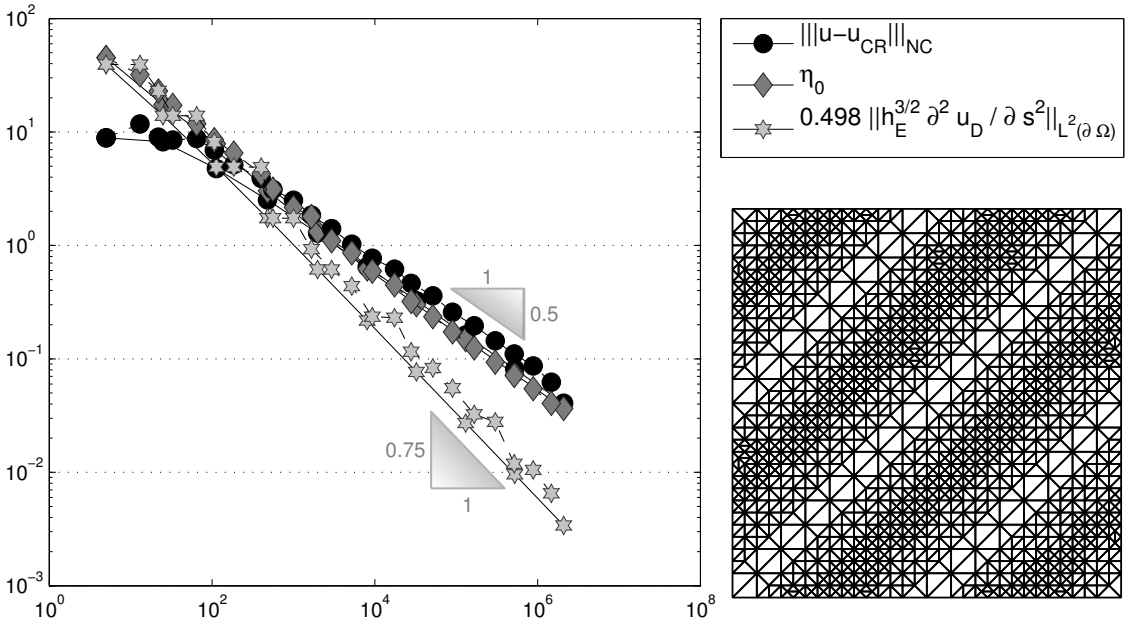

Figure 5.20: Convergence history for the energy error $|||u - u_{\mathrm{CR}}|||_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.7.2. The right image shows the adaptive mesh on level 13.

ndof	k=1	k=2	k=3	k=4	k=5	k=10
5	5.085	5.085	5.085	5.085	5.085	5.085
13	2.824	2.805	2.799	2.797	2.796	2.796
22	2.931	2.931	2.931	2.931	2.931	2.931
33	2.564	2.559	2.558	2.558	2.558	2.558
65	1.760	1.690	1.671	1.666	1.664	1.663
107	1.671	1.607	1.597	1.596	1.596	1.596
185	1.673	1.608	1.589	1.583	1.581	1.579
401	1.545	1.477	1.458	1.451	1.449	1.446
561	1.483	1.415	1.397	1.391	1.389	1.388
1001	1.393	1.314	1.294	1.287	1.285	1.283
1665	1.454	1.383	1.365	1.359	1.356	1.354
2953	1.319	1.240	1.221	1.215	1.213	1.212
5205	1.349	1.274	1.255	1.249	1.247	1.245
9329	1.302	1.224	1.206	1.200	1.199	1.197
17445	1.286	1.203	1.183	1.177	1.176	1.175
27889	1.260	1.178	1.159	1.153	1.152	1.150
51769	1.242	1.155	1.135	1.130	1.128	1.127

Table 5.2: Efficiency indices for η_{MP2} for different number of iterations k in Algorithm 5.1 for the minimisation of the sum of norms on adaptive meshes for the example from Subsection 5.7.2 with respect to the number of degrees of freedom (ndof).

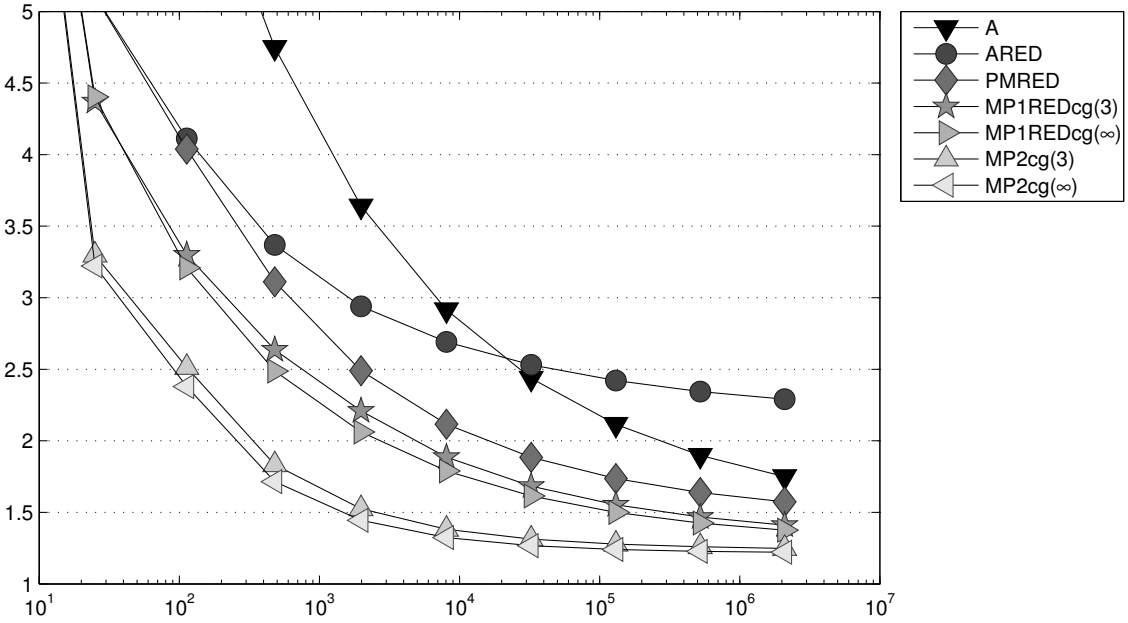

Figure 5.21: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.7.2.

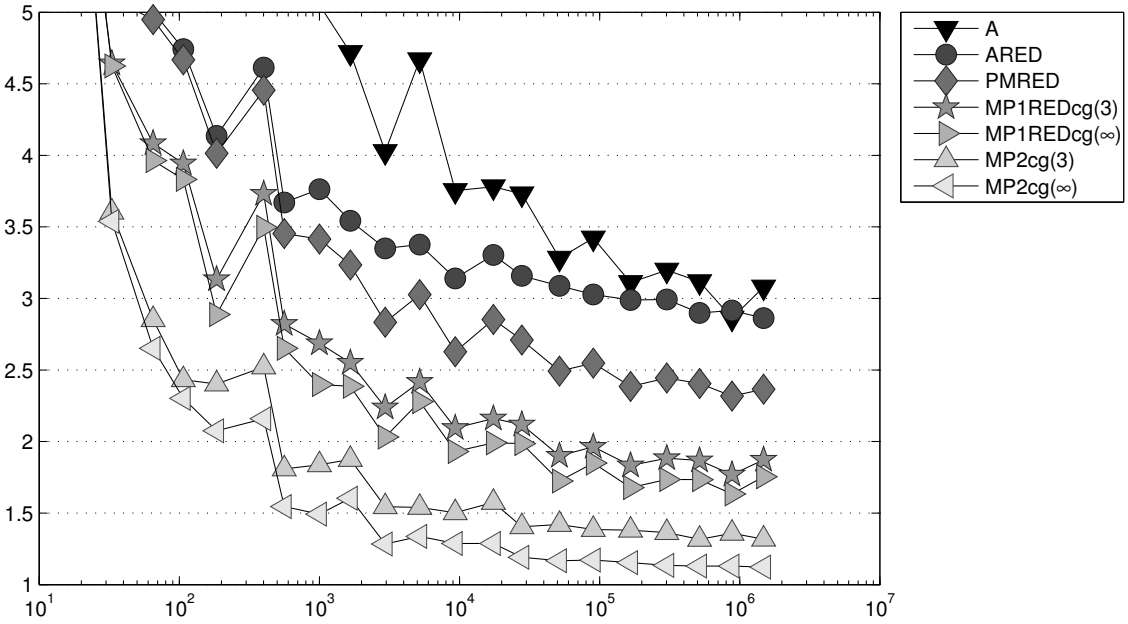

Figure 5.22: History of efficiency indices $\eta_{\mathrm{xyz}}/|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.7.2.

Table 5.2 shows that there is only little space for further improvement beyond $k = 3$ iterations. Therefore, $k = 3$ is a reasonable choice.

5.7.3 Another Smooth Example on Square Domain

In the third example with the data from Subsection 5.4.3 the error estimator η_{PMRED} with efficiency indices around 3 shows a very large improvement compared to η_{A} with efficiency indices between 4 and 5 (see Figures 5.25 and 5.26). The optimal error estimators based on a red-refinement yield only slightly better efficiency indices. The quadratic approximations $\eta_{\mathrm{MP2cg}(3)}$ and $\eta_{\mathrm{MP2cg}(\infty)}$ dominate the competition. Figure 5.23 shows the convergence history for the energy error and the overhead term due to the inhomogeneous Dirichlet data.

5.7.4 Colliding Flow

This subsection revisits the example from Subsection 5.4.4. The efficiency indices depicted in Figures 5.27 and 5.28 lead to similar conclusions as in the previous examples. The error estimator η_{MP2} and its pcg approximation $\eta_{\mathrm{MP2cg}(3)}$ with three iterations assume efficiency indices very close to 1, η_{A} assumes efficiency indices greater than 4. The error estimators η_{ARED} and η_{PMRED} assume indices around 3 and are almost as good as η_{MP1RED}. Figure 5.24 shows the convergence history for the energy error and the overhead term due to the inhomogeneous Dirichlet data.

5.7.5 Backward Facing Step

The last example concerns the backward facing step with the data from Subsection 5.4.5. The solution is unknown and so Figure 5.29 displays the convergence history for all error estimators under consideration and Figure 5.30 shows an adaptively generated mesh. The adaptive mesh refinement appears very reasonable and leads to the optimal convergence rate for all error estimators (and hence to optimal convergence of the smaller real energy error). Again, the quadratic error estimators $\eta_{\mathrm{MP2cg}(3)}$ and $\eta_{\mathrm{MP2cg}(\infty)}$ show the best results.

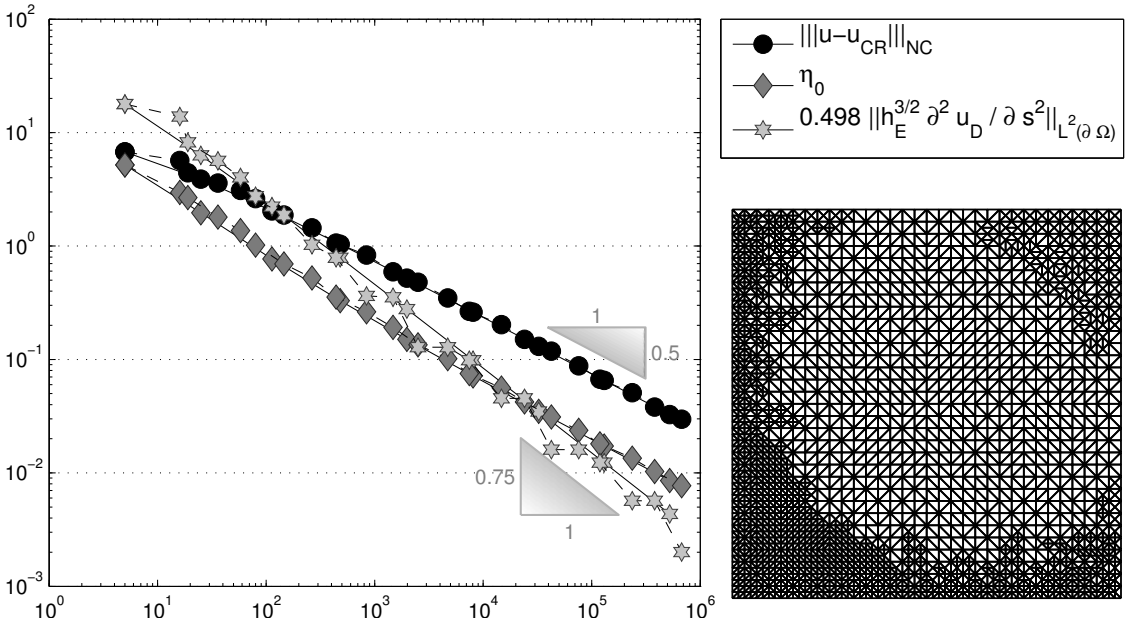

Figure 5.23: Convergence history for the energy error $\|\!|\!| u - u_{\mathrm{CR}} |\!|\!|_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.7.3. The right image shows the adaptive mesh on level 14.

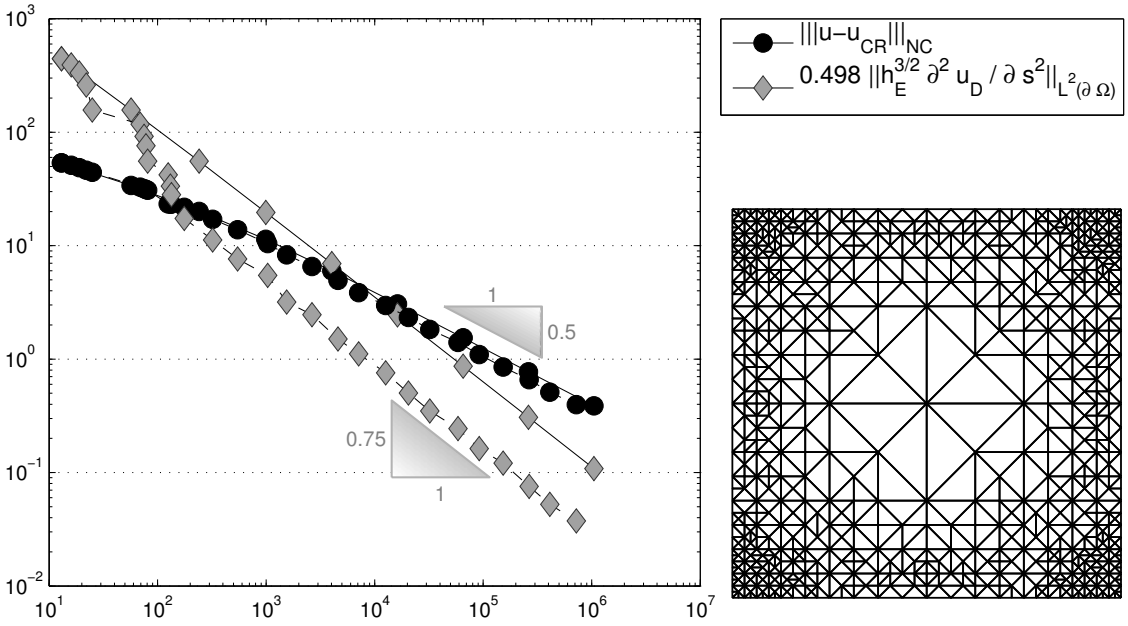

Figure 5.24: Convergence history for the energy error $\|\!|\!| u - u_{\mathrm{CR}} |\!|\!|_{\mathrm{NC}}$ and the Dirichlet error contribution $0.3652 \left\| h_{\mathcal{E}}^{3/2} \partial^2 u_D / \partial s^2 \right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.7.4. The right image shows the adaptive mesh on level 18.

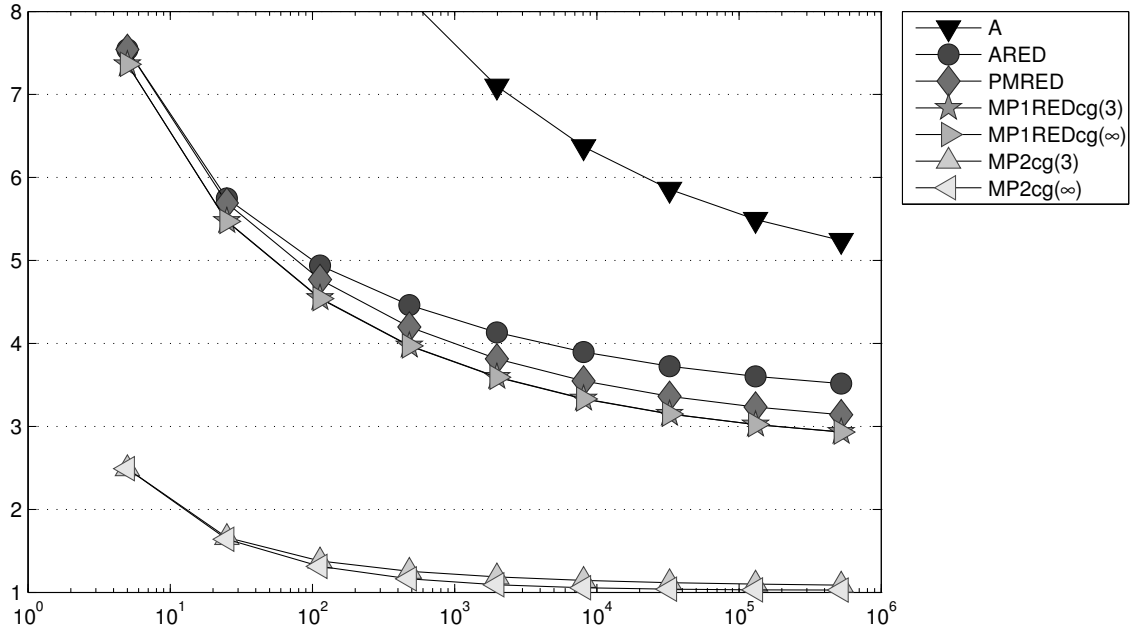

Figure 5.25: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.7.3.

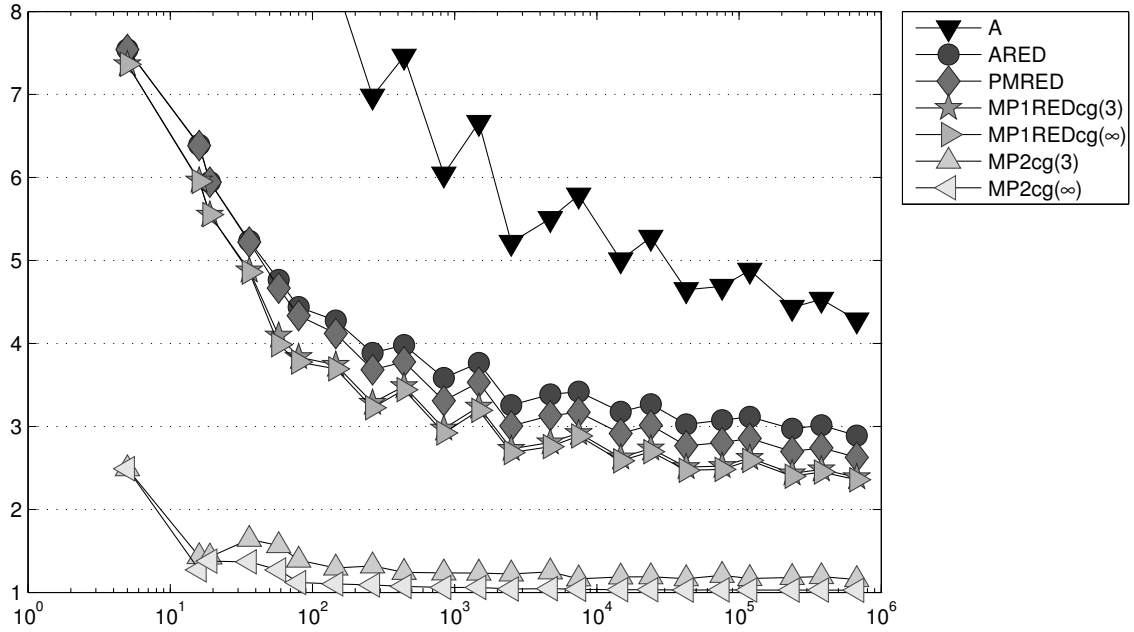

Figure 5.26: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.7.3.

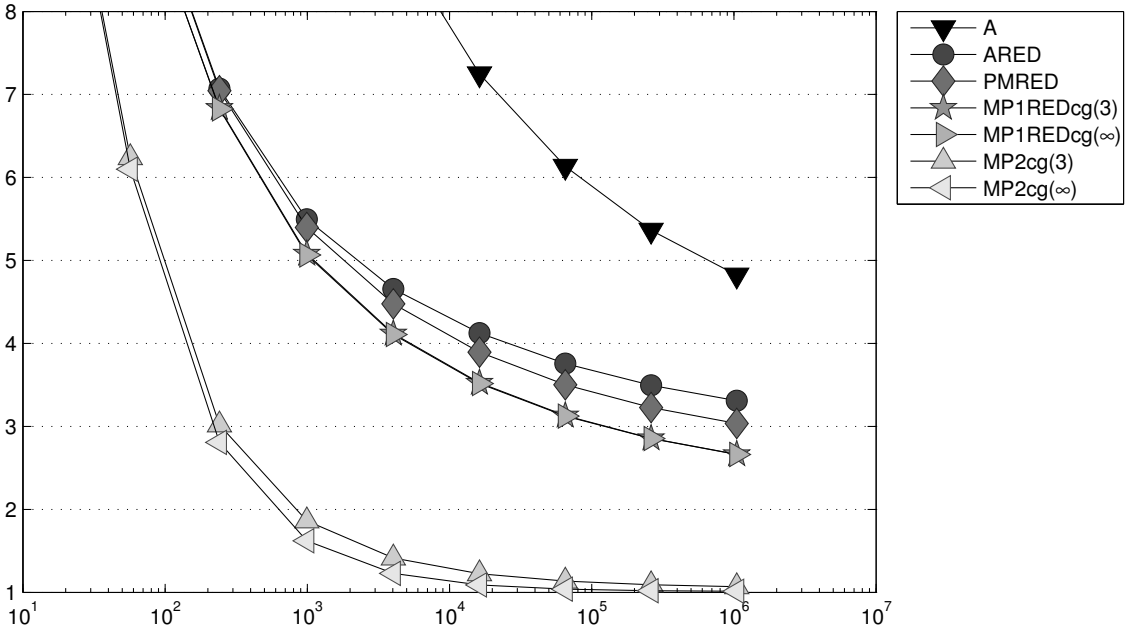

Figure 5.27: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 5.7.4.

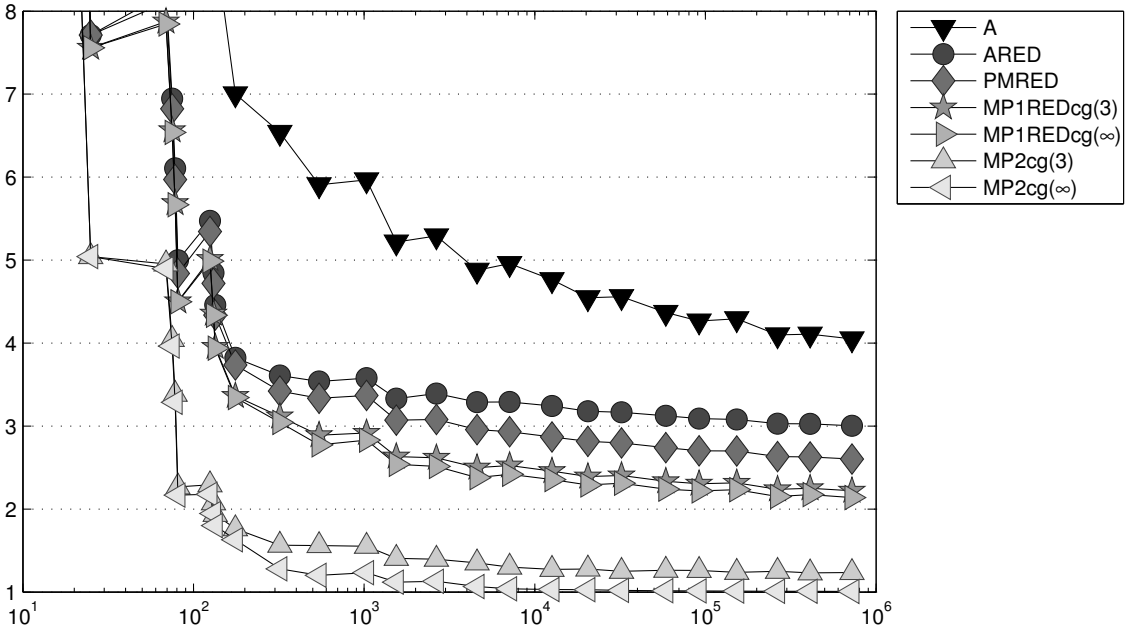

Figure 5.28: History of efficiency indices $\eta_{\mathrm{xyz}}/\,|||u-u_M|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 5.7.4.

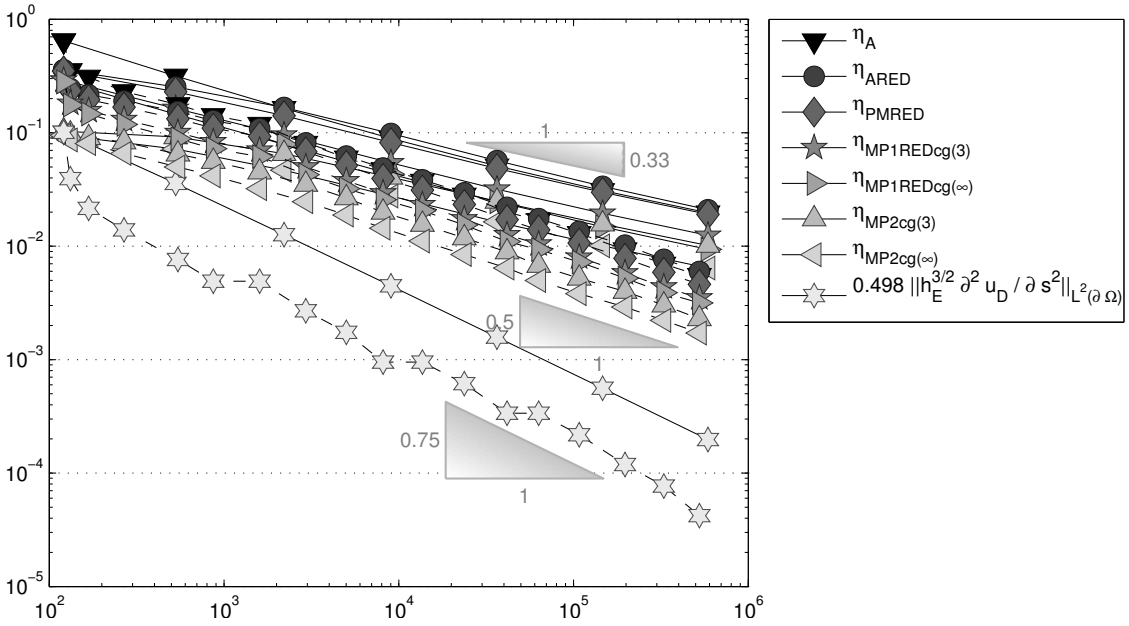

Figure 5.29: Convergence history for the error estimators η_{xyz} and the Dirichlet error contribution $0.3652\left\|h_{\mathcal{E}}^{3/2}\partial^2 u_D/\partial s^2\right\|_{L^2(\partial\Omega)}$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 5.7.5.

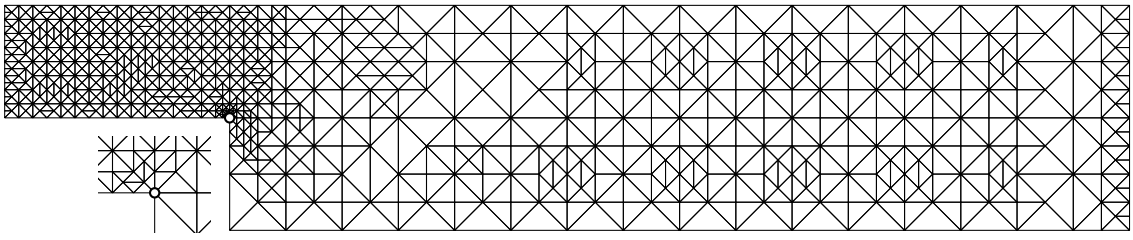

Figure 5.30: Adaptive mesh on level 8 in Subsection 5.7.5 and the neighbourhood of the singular point $(0,0)$ magnified by a factor 4.

6 Error Analysis for the Obstacle Problem

The obstacle problem is the model example for variational inequalities and connected to the study of minimal surfaces or capacitary potentials. This chapter reports on the results by Carstensen and Merdon (2012) and extends the analysis in some aspects that concern the overhead terms and inexact solve.

6.1 Setting

For a given obstacle function $\chi \in H^1(\Omega)$ with $\chi \leqslant u_D$ and Dirichlet boundary data $u_D \in H^1(\Gamma_D)$, the set of all admissible functions reads

$$\mathcal{K} := \left\{ v \in H^1(\Omega) \,\middle|\, \chi \leqslant v \text{ in } \Omega \text{ and } u_D = v \text{ along } \Gamma_D \right\}.$$

The obstacle problem seeks the minimiser of the energy E from (4.3) amongst $\mathcal{K}$, i.e.,

$$u = \operatorname*{argmin}_{v \in \mathcal{K}} E(v).$$

This is equivalent to the solve of the variational inequality

$$a(u, u - v) \leqslant F(u - v) \qquad \text{for all } v \in \mathcal{K}. \tag{6.1}$$

For details and other facts, e.g. unique existence and regularity results, confer to textbooks like (Kinderlehrer and Stampacchia, 1980). The residual

$$\varrho := F - a(u, \cdot) \in V^\star := H^1_D(\Omega)^\star \tag{6.2}$$

carries the contact information. To see this, consider the maximal open set $\mathcal{C}$ with $u = \chi$ on $\mathcal{C}$ and $\mathcal{D} := \bigcup_{\varepsilon > 0} B_\varepsilon$ with the maximal open set B_ε with $\chi + \varepsilon \leqslant u$ on B_ε. These sets describe the contact and the noncontact set and, under sufficient regularity assumptions on u and χ, it holds $\Delta u + f = 0$ in $\mathcal{D}$. This and (6.1) lead to

$$\begin{aligned}
&\varrho(v) \leqslant 0 && \text{for all } v \in V \text{ with } v \geqslant 0, \\
&\varrho(v) = \int_\Omega (f + \Delta\chi) v \,\mathrm{d}x && \text{for all } v \in C^\infty_C(\mathcal{C}) \cap V, \\
&\varrho(v) = 0 && \text{for all } V \in C^\infty_C(\mathcal{D}) \cap V.
\end{aligned} \tag{6.3}$$

This implies that ϱ locally has a Riesz representation Λ with

$$\Lambda|_{\mathcal{C}} := 0 \in L^2(\mathcal{C}) \quad \text{and} \quad \Lambda|_{\mathcal{D}} := f + \Delta\chi \in L^2(\mathcal{D}), \tag{6.4}$$

```
TolFun=eps^4;
TolPCG=0.01;
TolX=2.2204e-14;
MaxIter=10000;
ub=chi(c4n);
options = optimset('Display','off','MaxIter',MaxIter,'TolFun',TolFun,'TolX',TolX,'TolPCG',TolPCG);
y=quadprog(A(freeNodes,freeNodes),-b(freeNodes),[],[],[],[],ub(freeNodes),[],x0(freeNodes),options);
```

Listing 6.1: Listing for `solveP1Obstacle.m`

but Λ is not necessarily in L^2 on the remaining free boundary $\mathcal{F} := \Omega \setminus (\mathcal{C} \cup \mathcal{D})$.

6.2 Discretisation

The discretisation on a regular triangulation $\mathcal{T}$ involves the nodal interpolations $\chi_h := \sum_{z\in\mathcal{N}} \chi(z)\varphi_z$ of χ and $u_{D,h} := \sum_{z\in\mathcal{N}(\Gamma_D)} u_D(z)\varphi_z$ of the given Dirichlet data $u_D \in H^1(\Gamma_D)$. The discrete set of admissible functions reads

$$\mathcal{K}(\mathcal{T}) := \left\{ v_h \in P_1(\mathcal{T}) \cap C(\Omega) \,\middle|\, v_h = u_{D,h} \text{ along } \Gamma_D \text{ and } \chi_h \leqslant v_h \right\}$$

and is closed, non-void and convex. The finite element method for the obstacle set seeks $u_h \in \mathcal{K}(\mathcal{T})$ with

$$a(u_h, u_h - v_h) \leqslant F(u_h - v_h) \quad \text{for all } v_h \in \mathcal{K}(\mathcal{T}). \tag{6.5}$$

If $\chi \leqslant \chi_h$ (e.g. for convex functions χ), it follows $\mathcal{K}(\mathcal{T}) \subseteq \mathcal{K}$ and the problem is called a *conforming obstacle problem*. The `MATLAB` solver quadprog solves this problem by an iterative large-scale subspace trust-region method with linearised problems in each iteration that are solved by some pcg scheme. Listing 6.1 displays the used tolerances. Except `TolFun` all tolerances are set to the default values. However, undisplayed experiments reveal that one term in the guaranteed error bound reacts very sensible to the nonexact solve of this problem. That is why `TolFun` is chosen this small (notice that `eps^4=2.4309e-63`). The Matrix `A` and vector `b` are the same as for the Poisson model problem, since the same energy is minimised. The obstacle constraint is contained in the vector `ub` in the sense that `ub(freeNodes)` $\leqslant$ `y`. The vector `freeNodes` contains the node numbers of all free nodes in $\mathcal{M}$, i.e., all nodes that are not part of the Dirichlet boundary, and `y` contains the coefficients of their corresponding nodal basis functions.

The discrete counterpart of ϱ from (6.2) reads

$$\varrho_h := F - a(u_h, \cdot) \in V(\mathcal{T})^\star. \tag{6.6}$$

The following Lemma states the *discrete complementary conditions* that can be seen as an analogon to (6.3).

Lemma 6.2.1 (Discrete Complementary Conditions)**.** *It holds*

$$(u_h(z) - \chi_h(z))\varrho_h(\varphi_z) = 0 \quad \textit{and} \quad \varrho_h(\varphi_z) \leqslant 0 \quad \textit{for all free nodes } z \in \mathcal{M}.$$

Moreover, for any $w_D \in H^1(\Omega)$ with $w_D = u_D - u_{D,h}$ on Γ_D and $\chi - u_h \leqslant w_D$ in Ω, it holds

$$0 \leqslant \varrho(u - u_h - w_D).$$

Proof. The definitions in (6.6) and (6.5) show

$$0 \leqslant \varrho_h(u_h - v_h) \quad \text{for all } v_h \in \mathcal{K}(\mathcal{T}).$$

The choice $v_h = u_h + \varphi_z \in \mathcal{K}(\mathcal{T})$ yields $0 \leqslant \varrho_h(\varphi_z)$. The choice $v_h = u_h + (\chi_h(z) - u_h(z))\varphi_z \in \mathcal{K}(\mathcal{T})$ and $\chi_h(z) - u_h(z) \leqslant 0$ lead to

$$0 \leqslant (\chi_h(z) - u_h(z))\varrho_h(\varphi_z) \leqslant 0.$$

This implies the second assertion. The last assertion is a direct consequence of (6.2) and (6.1) for the test function $v = u_h + w_D \in \mathcal{K}$. □

This section concludes with some notation for a smooth presentation of the subsequent error analysis. The set of all triangles $T \in \mathcal{T}$ with contact of the discrete solution reads

$$\mathcal{T}_C := \left\{ T \in \mathcal{T} \,\middle|\, u_h = \chi_h \text{ on } T \right\}.$$

The set of all triangles $T \in \mathcal{T}(\Gamma_D)$ along the Dirichlet boundary Γ_D with contact of the discrete solution on the neighbourhood ω_T is denoted by

$$\mathcal{T}_{DC} := \left\{ T \in \mathcal{T}(\Gamma_D) \,\middle|\, u_h = \chi_h \text{ on } \omega_T \right\}.$$

The set of all triangles T in some layer between $\mathcal{T}_C$ and the non-contact set $\{x \in \Omega \,|\, \chi_h(x) < u_h(x)\}$ is denoted by

$$\mathcal{T}_i := \{T \in \mathcal{T} \,|\, \exists x, y \in \mathcal{N}(\overline{\omega}_T),\ \chi_h(x) = u_h(x) \text{ and } \chi_h(y) < u_h(y)\}.$$

Here $\mathcal{N}(\overline{\omega}_T) := \{z \in \mathcal{N} \,|\, z \in \overline{\omega}_T\}$ denotes all nodes in the element patch ω_T of the triangle T. For each element $T \in \mathcal{T}_i$, let $z_T \in \mathcal{N} \cap \overline{\omega}_T$ denote the (preferably interior) node with $\chi_h(z_T) = u_h(z_T)$. All elements $T \in \mathcal{T}_i$ with $z_T \in \Gamma_N$ form the set

$$\mathcal{T}_N := \{T \in \mathcal{T}_i \,|\, z_T \in \Gamma_N\}.$$

6.3 A Posteriori Error Analysis for Obstacle Problems

This section derives guaranteed upper bounds for the energy error via the methodology by Braess (2005) in the distinct realisation of Carstensen and Merdon (2012) with explicit constants and inhomogeneous boundary conditions.

6.3.1 Braess Methodology

The discrete complementary conditions in Lemma (6.2.1) convey that ϱ_h from (6.6) contains information about the contact zone $\{x \in \Omega \,|\, u_h(x) = \chi_h(x)\}$. The Braess

methodology incorporates this information and suggests some discrete contact force $\Lambda_h \in P_1(\mathcal{T}) \cap H^1(\Omega)$ such that

$$\int_\Omega \Lambda_h \varphi_z \,\mathrm{d}x = \varrho_h(\varphi_z) \quad \text{for all } z \in \mathcal{M}.$$

However, this linear problem is under-determined as it only concerns the free nodes $z \in \mathcal{M}$ and it is not so clear how to incorporate the Dirichlet boundary nodes. The evaluation $\varrho_h(\varphi_z)$ by (6.6) possibly makes no sense for $z \in \mathcal{N}(\Gamma_D)$. On the other hand, Veeser (2001, line 25 on page 153) observed that $\varrho_h(\varphi_z) \coloneqq 0$ for $z \in \mathcal{N}(\Gamma_D)$ may lead to large refinement indicators. An extrapolation of the known contact information for the free nodes to the fixed Dirichlet boundary nodes from Carstensen and Merdon (2012) circumvents that problem.

The extrapolation employs a mapping $\zeta : \mathcal{N} \to \mathcal{M}$, which is the identity restricted to $\mathcal{M}$, i.e. $\zeta(z) = z$ for $z \in \mathcal{M}$, but maps a Dirichlet boundary node $z \in \mathcal{N}(\Gamma_D)$ to some neighbouring free node $\zeta(z) \in \mathcal{M}$. Later, Section 6.3.3 requires one free node in every element $T \in \mathcal{T}$ and $\zeta(z) \in \overline{\omega}_T$ for all $z \in \mathcal{N}(T)$. However, this is not needed for the proof of reliability in Section 6.3.2.

The mapping ζ defines a partition of $\mathcal{N}$ into $|\mathcal{M}|$ preimages

$$\zeta^{-1}(z) \coloneqq \{y \in \mathcal{N} \,|\, \zeta(y) = z\} \text{ for each } z \in \mathcal{M}.$$

The sum of basis functions for the nodes that belong to the same preimage defines a new partition of unity as in (Carstensen, 1999),

$$\psi_z \coloneqq \sum_{y \in \zeta^{-1}(z)} \varphi_y \in P_1(\mathcal{T}) \cap C(\Omega) \quad \text{for all } z \in \mathcal{M}. \tag{6.7}$$

Finally, for each $z \in \mathcal{N}$, set

$$\varrho_h^\star(\varphi_z) \coloneqq \begin{cases} 0 & \text{if } \chi(z) < u_h(z), \\ \varrho_h(\varphi_{\zeta(z)}) \frac{\int_\Omega \varphi_z \,\mathrm{d}x}{\int_\Omega \varphi_{\zeta(z)} \,\mathrm{d}x} & \text{else.} \end{cases} \tag{6.8}$$

A direct consequence of (6.7) and Lemma (6.2.1) is

$$\sum_{z \in \mathcal{N}} \frac{\varrho_h^\star(\varphi_z)}{\int_\Omega \varphi_z \,\mathrm{d}x} \varphi_z = \sum_{z \in \mathcal{M}} \frac{\varrho_h(\varphi_z)}{\int_\Omega \varphi_z \,\mathrm{d}x} \psi_z \leqslant 0. \tag{6.9}$$

Eventually, $\Lambda_h \in \mathcal{P}_1(\mathcal{T}) \cap C(\Omega)$ denotes the nonpositive Riesz representation of $\varrho^\star$ in $P_1(\mathcal{T}) \cap C(\Omega)$, i.e.,

$$\int_\Omega \Lambda_h \varphi_z \,\mathrm{d}x = \varrho_h^\star(\varphi_z) \quad \text{for all } z \in \mathcal{N}. \tag{6.10}$$

Note that $\varrho_h^\star$ equals ϱ_h from Bartels and Carstensen (2004).

Listing 6.2 computes the residuals $\varrho_h(\varphi_z)$ for all nodes $z \in \mathcal{N}$ and stores the values in the vector `res4n`. The computation in Line 5 employs the stiffness matrix `A` and the

```
function [res4n,x4LM] = OBS_DiscreteLagrangeMultiplier(x,A,b,chi,c4n,n4e,n4sDb)
DbNodes = unique(n4sDb);
freeNodes = setdiff(1:size(x,1),DbNodes);
res4n = zeros(size(x));
res4n(freeNodes) = A(freeNodes,freeNodes)*x(freeNodes)-b(freeNodes);
free4Db = zeros(length(DbNodes),1);
c4FreeNodes = c4n(freeNodes,:);
for j=1:length(DbNodes)
    curNode = DbNodes(j);
    difference(:,1) = c4FreeNodes(:,1) - c4n(curNode,1);
    difference(:,2) = c4FreeNodes(:,2) - c4n(curNode,2);
    differences = sum(difference.*difference,2);
    [~,I] = min(differences);
    free4Db(j) = freeNodes(I(1));
end
area4n = computeArea4n(c4n,n4e);
res4n(DbNodes) = res4n(free4Db).*(area4n(DbNodes)./area4n(free4Db));
difference = x(DbNodes) - chi(c4n(DbNodes,:));
res4n(DbNodes(find(difference > eps))) = 0;
area4e = computeArea4e(c4n,n4e);
nrElems = size(n4e,1);
S = zeros(3,3,nrElems);
mama = reshape(reshape((ones(3)+eye(3))/12,[9 1])*ones(1,nrElems),[3 3 nrElems]);
S(1:3,1:3,:) = matMul(mama,permute(area4e,[2 3 1]));
n4eT = n4e';
I = repmat(n4eT(:),1,size(n4eT,1))';
J = repmat(n4e,1,size(n4eT,1))';
A = sparse(I(:),J(:),S(:));
x4LM = A\res4n;
```

Listing 6.2: Listing for `OBS_DiscreteLagrangeMultiplier.m`

right-hand side `b` of the solver `solveP1Obstacle`. Lines 6–15 associate each Dirichlet node in `DbNodes` with the closest free node, which is stored in `free4Db`. Then the corresponding entries in `res4n(DbNodes)` are changed in Lines 17–19 to satisfy (6.8). The remaining Lines 20–29 compute the coefficients `x4LM` of the Riesz representation Λ_h of $\varrho_h^\star$.

Remark 6.3.1. *Veeser (2001) suggests the alternative boundary modification*

$$\varrho_h^\star(\varphi_z) \coloneqq \begin{cases} 0 & \text{if } \chi(z) < u_h(z), \\ \min\left\{0, F(\varphi_z) - a(u_h,\varphi_z) + \int_{\Gamma_D} \varphi_z \nabla u_h \cdot \nu \,\mathrm{d}s\right\} & \text{else.} \end{cases}$$

This resembles the original definition of the residual ϱ_h from (6.6) *with an additional boundary term that appears after an integration by parts in $a(u_h,\varphi_z)$.*

6.3.2 Guaranteed Upper Error Bounds

This sections designs guaranteed upper bounds (GUB) for the energy error $|||u - u_h|||$. The analysis employs the quasi interpolation operator from Carstensen (1999) that reads

$$J(v) \coloneqq \sum_{z\in\mathcal{N}} J_z(v)\varphi_z \quad \text{with} \quad J_z(v) \coloneqq \int_{\omega_z} v\varphi_z \,\mathrm{d}x \Big/ \int_{\omega_z} \varphi_z \,\mathrm{d}x \quad \text{for any } v \in L^2(\Omega). \tag{6.11}$$

The comparison between $J(v)$ and v spawns the node patch oscillations

$$\operatorname{osc}(v,\mathcal{N}) \coloneqq \left(\sum_{z\in\mathcal{N}} \|h_z(v-J_z(v))\varphi_z\|^2_{L^2(\omega_z)}\right)^{1/2}. \tag{6.12}$$

Here $v_{\omega_z} \coloneqq \int_{\omega_z} v\,\mathrm{d}x/|\omega_z| \in P_0(\omega_z)$ denotes the integral mean of v on the patch ω_z. Note that these oscillations include φ_z in contrast to the ones in Carstensen and Merdon (2012). This allows sharper guaranteed upper bounds.

Lemma 6.3.2 (Properties of J)**.** *For $f \in L^2(\Omega)$ and $v \in H^1(\Omega)$, it holds*

(a) $\int_\Omega (f-Jf)v\,\mathrm{d}x = \int_\Omega f(v-Jv)\,\mathrm{d}x$,
(b) $\int_{\omega_z}(f-J_z(f))\varphi_z\,\mathrm{d}x = 0$ *for all $z\in\mathcal{N}$*,
(c) $\int_\Omega (f-Jf)v\,\mathrm{d}x \leqslant (n+1)^{1/2}\max_{z\in\mathcal{N}} C_{\mathrm{P}}(\omega_z)\ |||v||| \operatorname{osc}(f,\mathcal{N})$ *with* $\operatorname{osc}(f,\mathcal{N})$ *from* (6.12).

Proof. The proof follows Carstensen (1999) and only the proof of the third property is repeated here to calculate the explicit constant. The second property and $v_{\omega_z} = \fint_{\omega_z} v\,\mathrm{d}x$ lead to

$$\begin{aligned}\int_\Omega (f-Jf)v\,\mathrm{d}x &= \sum_{z\in\mathcal{N}}\int_{\omega_z}(f-J_z(f))\varphi_z(v-v_{\omega_z})\,\mathrm{d}x\\ &\leqslant \sum_{z\in\mathcal{N}} \|h_z(f-J_z(f))\varphi_z\|_{L^2(\omega_z)}\left\|h_z^{-1}(v-v_{\omega_z})\right\|_{L^2(\omega_z)}.\end{aligned}$$

A Cauchy inequality in $\mathbb{R}^{|\mathcal{N}|}$ and local Poincaré inequalities on ω_z for all $z\in\mathcal{N}$ yield

$$\begin{aligned}\int_\Omega (f-Jf)v\,\mathrm{d}x &\leqslant \left(\sum_{z\in\mathcal{N}}\|h_z(f-J_z(f))\varphi_z\|^2_{L^2(\omega_z)}\right)^{1/2}\left(\sum_{z\in\mathcal{N}}\left\|h_z^{-1}(v-v_{\omega_z})\right\|^2_{L^2(\omega_z)}\right)^{1/2}\\ &\leqslant \operatorname{osc}(f,\mathcal{N})\left(\sum_{z\in\mathcal{N}} C_{\mathrm{P}}(\omega_z)^2\,\|\nabla v\|^2_{L^2(\omega_z)}\right)^{1/2}.\end{aligned}$$

An overlap argument (each element is part of $n+1$ node patches) concludes the proof. □

Theorem 6.3.3 below proves reliability for some guaranteed upper bound that consists of computable terms with known and explicit constants.

Theorem 6.3.3. *Let u denote the exact solution of* (6.1) *with ϱ from* (6.2) *and let u_h denote the discrete solution of* (6.5) *with the associated $\varrho_h^\star$ from* (6.8) *and its Riesz representation Λ_h from* (6.10)*. For $\chi \leqslant \chi_h$ and $w_D \in H^1(\Omega)$ with $w_D = u_D - u_{D,h}$ on Γ_D and $\chi - u_h \leqslant w_D$ in Ω, it holds*

$$\begin{aligned}|||u-u_h||| &\leqslant \alpha/2 + \sqrt{\alpha^2/4+\beta} + |||w_D|||,\\ |||\varrho-\varrho_h^\star|||_\star &\leqslant |||\mathrm{Res}_{\mathrm{aux}}|||_\star + |||u-u_h|||\end{aligned}$$

with the auxiliary residual

$$\mathrm{Res}_{\mathrm{aux}}(v) \coloneqq \int_\Omega (f - \Lambda_h) v \,\mathrm{d}x + \int_{\Gamma_N} g v \,\mathrm{d}s - \int_\Omega \nabla u_h \cdot \nabla v \,\mathrm{d}x \quad \textit{for } v \in V,$$

and the quantities

$$\alpha \coloneqq |||\mathrm{Res}_{\mathrm{aux}}|||_\star + |||\Lambda_h - J\Lambda_h|||_\star + |||w_D||| \quad \textit{and} \quad \beta \coloneqq \int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x.$$

Proof. The proof is similar to the proof of Lemma 3.1 from Braess (2005) for $\sigma_h^+ \coloneqq -J\Lambda_h$, but is repeated here for the more general case and a slightly sharper upper bound. Let w denote the exact solution of the auxiliary Poisson problem

$$\Delta w + f - \Lambda_h = 0 \text{ in } \Omega, \quad w = u_{D,h} \text{ along } \Gamma_D \quad \text{and} \quad \nabla w \cdot \nu = g \text{ along } \Gamma_N. \tag{6.13}$$

Note that, by (6.10), u_h is the discrete solution of this auxiliary Problem with exact Dirichlet boundary data. Hence, it holds $|||\mathrm{Res}_{\mathrm{aux}}|||_\star = |||w - u_h|||$. The definitions of Λ_h from (6.10) and ϱ from (6.2) and the application of J from (6.11) show, for any $v \in V$,

$$\begin{aligned} a(u - w, v) = \int_\Omega v\Lambda_h \,\mathrm{d}x - \varrho(v) &= \int_\Omega vJ\Lambda_h \,\mathrm{d}x - \varrho(v) + \int_\Omega v(\Lambda_h - J\Lambda_h) \,\mathrm{d}x \\ &\leqslant \int_\Omega vJ\Lambda_h \,\mathrm{d}x - \varrho(v) + |||\Lambda_h - J\Lambda_h|||_\star . \end{aligned}$$

For $v \coloneqq u - u_h - w_D = u - u_h - w_D \in V$, it holds

$$\begin{aligned} &\int_\Omega (u - u_h - w_D) J\Lambda_h \,\mathrm{d}x - \varrho(u - u_h - w_D) \\ &= \int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x - \int_\Omega (\chi - u) J\Lambda_h \,\mathrm{d}x - \varrho(u - u_h - w_D). \end{aligned}$$

Lemma 6.3.2, $\chi - u \leqslant 0$, and $J\Lambda_h \leqslant 0$ from yield

$$0 \leqslant \int_\Omega (\chi - u) J\Lambda_h \,\mathrm{d}x \quad \text{and} \quad 0 \leqslant \varrho(u - u_h - w_D).$$

Hence,

$$\int_\Omega (u - u_h - w_D) J\Lambda_h \,\mathrm{d}x - \varrho(u - u_h - w_D) \leqslant \int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x.$$

The previous results and some algebra lead to

$$\begin{aligned} &|||u - u_h - w_D|||^2 \\ &= a(u - w, u - u_h - w_D) + a(w - u_h, u - u_h - w_D) - a(w_D, u - u_h - w_D) \\ &\leqslant \Big(|||w - u_h||| + |||\Lambda_h - J\Lambda_h|||_* + |||w_D||| \Big) |||u - u_h - w_D||| + \int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x. \end{aligned}$$

This is an inequality of the form $x^2 \leqslant \alpha x + \beta$ and elementary computations lead to

$$0 \leqslant x \leqslant \alpha/2 + \sqrt{\alpha^2/4 + \beta}.$$

This and the triangle inequality $|||u - u_h||| \leqslant |||u - u_h - w_D||| + |||w_D|||$ prove the first assertion. Furthermore, (6.2) and (6.6) yield

$$\varrho(v) - \varrho_h^\star(v) = a(u - w, v) \leqslant |||u - w|||\,|||v||| \text{ for all } v \in V.$$

Hence,

$$|||\varrho - \varrho_h^\star|||_\star \leqslant |||u - w|||\,.$$

The triangle inequality and $|||\mathrm{Res}_{\mathrm{aux}}|||_\star = |||w - u_h|||$ conclude the proof for the second assertion. □

This section concludes with some remarks.

Remark 6.3.4. *(a) Lemma 6.3.2.(c) shows*

$$|||\Lambda_h - J\Lambda_h|||_\star \leqslant (n+1)^{1/2} \max_{z \in \mathcal{N}} C_{\mathrm{P}}(\omega_z)\,\mathrm{osc}(\Lambda_h, \mathcal{N}).$$

For convex patches ω_z the constant $(n+1)^{1/2}/\pi$ is well below 1 and so 1 is used for the numerical experiments in Section 6.4, confer to Veeser and Verfürth (2012) for bounds of Poincaré constants $C_{\mathrm{P}}(\omega_z)$ on finite element stars.
(b) Since the discrete problem cannot be solved exactly, the nonpositivity condition of $J\Lambda_h$ might be violated. However, there is a trick to modify the analysis to cope with inexact solutions. To do so, we replace $J\Lambda_h$ by $\min(J\Lambda_h, 0)$ and end up with the modified quantities

$$\alpha := |||\mathrm{Res}_{\mathrm{aux}}|||_\star + |||\Lambda_h - \min(J\Lambda_h, 0)|||_\star + |||w_D||| \quad \textit{and}$$
$$\beta := \int_\Omega (\chi - u_h - w_D)\min(J\Lambda_h, 0)\,\mathrm{d}x.$$

A triangle inequality and a Friedrichs inequality yield

$$|||\Lambda_h - \min(J\Lambda_h, 0)|||_\star \leqslant |||\Lambda_h - J\Lambda_h|||_\star + C_F\,\|J\Lambda_h - \min(J\Lambda_h, 0)\|_{L^2(\Omega)}\,.$$

This still allows to exploit the orthogonality property of J as in Remark 6.3.4.(a). The latter term is zero for exact solve of the discrete problem and might even be used for adaptive control of the tolerances of the iterative solver. However, such experiments are out of the scope of this thesis.
(c) The term $\int_\Omega (\chi - u_h - w_D)J\Lambda_h\,\mathrm{d}x$ can be evaluated exactly (up to quadrature errors). In case $\chi = \chi_h$, it only contributes on a layer between the discrete contact zone and the discrete non-contact zone $\{T \in \mathcal{T} \,|\, \exists z, y \in \mathcal{N}(T),\ \chi(z) < u_h(z)\ \&\ \widehat{\sigma}_h(\varphi_y) < 0\}$.
(d) The properties of w_D from Theorem 4.2.2 guarantee that the term $|||w_D|||$ is bounded by the higher-order term

$$|||w_D||| \leqslant C_{\mathrm{D}}(\mathcal{E}(\Gamma_D))\left\|h_{\mathcal{E}}^{3/2}\partial_{\mathcal{E}}^2 u_D/\partial \mathrm{s}^2\right\|_{L^2(\Gamma_D)}.$$

Furthermore w_D contributes to $\int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x$ only on elements with contact near the boundary. For homogeneous Dirichlet data, $w_D \equiv 0$.
(e) In case of a nonaffine obstacle $\chi_h \neq \chi \notin P_1(\mathcal{T})$, the test function $u_h - w_D$ might not be an admissible function in $\mathcal{K}$. To resolve this matter, one can replace w_D by $w_D - \min\{0, u_h - \chi\}$ as known from Falk (1974).
(f) Theorem 6.3.3 implies the reliable upper bound

$$|||\varrho - \varrho_h^\star|||_* \leqslant 2\,|||\mathrm{Res}_{\mathrm{aux}}|||_\star + |||\Lambda_h - J\Lambda_h|||_\star + \left(\int_\Omega (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x\right)^{1/2} + 2\,|||w_D|||\,.$$

6.3.3 Efficiency

Theorem 6.3.3 above showed reliability

$$|||u - u_h||| + |||\varrho - \varrho_h^\star|||_\star \leqslant \mathrm{GUB}$$

for the global upper bound

$$\mathrm{GUB} \coloneqq 3\,|||\mathrm{Res}_{\mathrm{aux}}|||_\star + 2\,|||\Lambda_h - J\Lambda_h|||_\star + 2\left(\int_\Omega (\chi - u_h - w_D) J\Lambda_h dx\right)^{1/2} + 4\,|||w_D|||\,.$$

This subsection proves the converse up to perturbation terms like oscillations. Recall the definitions of $\mathcal{T}_i$, $\mathcal{T}_{DC}$ and $\mathcal{T}_N$ from the end of Section 6.2. The proof of efficiency also employs the following Lemma from Bartels and Carstensen (2004).

Lemma 6.3.5 (Lemma 8 of Bartels and Carstensen (2004))**.** *Let $z \in \mathcal{N}$ be either an interior point of Ω or a nonconvex boundary point (so convex corner, in particular points on straight line segments are excluded). Suppose $T \in \mathcal{T}$, $\omega_T \coloneqq \left\{\sum_{z\in\mathcal{N}(T)} \varphi_z > 0\right\}$ with $z \in \overline{\omega}_T$ and set $\widehat{\Omega}_z \coloneqq \left\{x \in \Omega \,\middle|\, \psi_z(x) > 0\right\} \cup \omega_T$ with respect to the partition of unity functions ψ from* (6.7)*. Let $w_h \in \mathcal{P}_1(\mathcal{T}) \cap C(\Omega)$ satisfy $w_h(z) = 0$ and $0 \leqslant w_h$ on $\widehat{\Omega}_z$. Then, it holds*

$$\|w_h\|_{L^2(\widehat{\Omega}_z)} \lesssim h_z \min_{q_z \in \mathcal{P}_1(\mathcal{T}(\widehat{\Omega}_z);\mathbb{R}^2)\cap C(\widehat{\Omega}_z;\mathbb{R}^2)} \|\nabla w_h - q_z\|_{L^2(\widehat{\Omega}_z)}\,.$$

Theorem 6.3.6. *For an affine obstacle $\chi_h \equiv \chi \in \mathcal{P}_1(\Omega)$ and $f \in H^1(\Omega)$, it holds*

$$\begin{aligned}\mathrm{GUB} \lesssim\ & |||u - u_h||| + |||\varrho - \varrho_h^\star|||_\star + |||\varrho \circ (\mathbb{I} - J)|||_\star + \mathrm{osc}(f, \mathcal{T}) + \mathrm{osc}(g, \mathcal{E}(\Gamma_N)) \\ & + \left(\sum_{T\in\mathcal{T}_i\setminus\mathcal{T}_C} \left\|h_{\mathcal{T}}^2 \nabla f\right\|_{L^2(\omega_T)}^2\right)^{1/2} + \left(\sum_{T\in\mathcal{T}_{DC}} \|\nabla w_D\|_{L^2(T)} \left\|h_{\mathcal{T}} f\right\|_{L^2(T)}\right)^{1/2} \\ & + \left(\sum_{T\in\mathcal{T}_N} \|\nabla(u - \chi)\|_{L^2(T)}^2\right)^{1/2} + |||w_D|||\,.\end{aligned}$$

Proof. The proof consists of seven steps.
Step 1. It holds $|||\mathrm{Res}_{\mathrm{aux}}|||_\star = |||w - u_h||| \leqslant |||u - u_h||| + \left|||\varrho - \varrho_h^\star|||_\star\right.$.

Proof of Step 1. The definitions of ϱ, $\varrho_h^\star$ and w from (6.13) and a Hölder inequality show

$$\begin{aligned}\|\!|w-u_h|\!\|^2 &= a(w-u_h,w-u_h) = a(w-u,w-u_h)+a(u-u_h,w-u_h)\\ &= \varrho(w-u_h)-\varrho_h^\star(w-u_h)+a(u-u_h,w-u_h)\\ &\leqslant \left(\|\!|\varrho-\varrho_h^\star|\!\|_\star + \|\!|u-u_h|\!\|\right)\|\!|w-u_h|\!\|\,.\end{aligned}$$

This concludes the proof of Step 1.

Step 2. It holds $\|\!|J\Lambda_h-\Lambda_h|\!\|_\star \lesssim \|\!|\varrho-\varrho_h^\star|\!\|_\star + \|\!|\varrho\circ(\mathbb{I}-J)|\!\|_\star$.
Proof of Step 2. For any $v\in V$, the definition of Λ_h and the properties of J of Lemma 6.3.2 lead to

$$\begin{aligned}\int_\Omega(\Lambda_h-J\Lambda_h)v\,\mathrm{d}x &= \varrho_h^\star(v-Jv)-\Lambda(v-Jv)+\Lambda(v-Jv)\\ &\lesssim \|\!|v|\!\|\left(\|\!|\varrho-\varrho_h^\star|\!\|_\star+\|\!|\varrho\circ(\mathbb{I}-J)|\!\|_\star\right).\end{aligned}$$

This concludes the proof of Step 2.

Step 3. It holds

$$\begin{aligned}&\int_\Omega(\chi-u_h-w_D)J\Lambda_h\,\mathrm{d}x\\ &\quad\lesssim \sum_{T\in\mathcal{T}_{DC}} h_T\,\|\nabla w_D\|_{L^2(T)}\,\|J\Lambda_h\|_{L^2(T)} + \sum_{T\in\mathcal{T}_i\backslash\mathcal{T}_{DC}} h_T^2\,\|\nabla w_D\|_{L^2(T)}\,\|\nabla(J\Lambda_h)\|_{L^2(\omega_T)}\\ &\quad+\sum_{T\in\mathcal{T}_i\backslash\mathcal{T}_C} h_T^2\,\|\nabla(J\Lambda_h)\|_{L^2(\omega_T)}\min_{q_z\in\mathcal{P}_1(\mathcal{T}(\widehat{\Omega}_{z_T};\mathbb{R}^2))\cap C(\widehat{\Omega}_{z_T};\mathbb{R}^2)}\|\nabla(\chi-u_h)-q_z\|_{L^2(\widehat{\Omega}_{z_T})}\,.\end{aligned}$$

Proof of Step 3. The integral $\int_\Omega(\chi-u_h-w_D)J\Lambda_h\,\mathrm{d}x$ is analysed for each $T\in\mathcal{T}$. In case that $\chi<u_h$ on $\overline{\omega}_T$, Lemma 6.2.1 yields

$$J\Lambda_h=\sum_{z\in\mathcal{N}(T)}\varphi_z\varrho_h^\star(\varphi_z)/\int_\Omega\varphi_z\,\mathrm{d}x=0\quad\text{on } T.$$

For $T\in\mathcal{T}_i\backslash\mathcal{T}_C$ with $|\partial T\cap\Gamma_D|=0$, it holds $w_D=0$ on T and $(u_h-\chi)(z_T)=0$ for some $z_T\in\mathcal{N}(\overline{\omega}_T)$. Furthermore, there exists some $y_T\in\mathcal{N}(T)$ with $\chi(y_T)<u_h(y_T)$. Lemma 6.2.1 and (6.8) yield

$$J\Lambda_h(y_T)=\varrho_h^\star(\varphi_{y_T})/\int\varphi_{y_T}\,\mathrm{d}x=0$$

and a discrete Friedrichs inequality shows

$$\|J\Lambda_h\|_{L^2(T)}\lesssim h_T\,\|\nabla(J\Lambda_h)\|_{L^2(\omega_T)}\,. \tag{6.14}$$

If $z_T\in\mathcal{M}$, (6.14) and Lemma 6.3.5 yield

$$\int_T(\chi-u_h)J\Lambda_h\,\mathrm{d}x\leqslant\|\chi-u_h\|_{L^2(T)}\,\|J\Lambda_h\|_{L^2(T)}$$

$$\lesssim h_T^2 \|\nabla(J\Lambda_h)\|_{L^2(\omega_T)} \min_{q_z \in \mathcal{P}_1(\mathcal{T}(\hat{\Omega}_{z_T};\mathbb{R}^2))\cap C(\hat{\Omega}_{z_T};\mathbb{R}^2)} \|\nabla(\chi - u_h) - q_z\|_{L^2(\hat{\Omega}_{z_T})}.$$

If $z_T \in \Gamma_D$, Lemma 6.3.5 is not applicable. However, this case is insignificant for the following reason. Since z_T is chosen preferably as an interior node, $z_T \in \Gamma_D$ implies $\mathcal{N}(\overline{\omega}_z) \cap \{u_h = \chi_h\} \subseteq \mathcal{N}(\Gamma_D)$. Hence, $\varrho_h^\star(y) = 0$ for all $y \in \mathcal{N}(T)$. Consequently, $J\Lambda_h = 0$ on T. In case that the isolated contact node z_T belongs to a convex corner or a straight-line segment of Γ_N, two applications of (6.14) yield

$$\int_T (\chi - u_h) J\Lambda_h \,\mathrm{d}x \leqslant h_T^2 \|\nabla(\chi - u_h)\|_{L^2(T)} \|\nabla J\Lambda_h\|_{L^2(T)}.$$

In fact, free nodes on convex corners lead to exceptional situations in some second-order positive approximation (Nochetto and Wahlbin, 2002).

For $T \in \mathcal{T}$ with $|\partial T \cap \Gamma_D| > 0$ the integral equals

$$\begin{aligned}\int_T (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x &= \int_T (\chi - u_h) J\Lambda_h \,\mathrm{d}x - \int_T w_D J\Lambda_h \,\mathrm{d}x \\ &\leqslant \|\chi - u_h\|_{L^2(T)} \|J\Lambda_h\|_{L^2(T)} + \|w_D\|_{L^2(T)} \|J\Lambda_h\|_{L^2(T)}.\end{aligned}$$

Since $w_D = 0$ on $\partial T \backslash \Gamma_D$, a Friedrichs inequality shows

$$\int_T (\chi - u_h - w_D) J\Lambda_h \,\mathrm{d}x \lesssim \|\chi - u_h\|_{L^2(T)} \|J\Lambda_h\|_{L^2(T)} + h_T \|\nabla w_D\|_{L^2(T)} \|J\Lambda_h\|_{L^2(T)}.$$

The first summand vanishes if $u_h = \chi$ on T or $\chi < u_h$ on ω_T. Otherwise, it holds $T \in \mathcal{T}_i$ and Lemma 6.3.5 leads, for $z = z_T \in \mathcal{N}(\Omega)$, to

$$\|\chi - u_h\|_{L^2(\hat{\Omega}_z)} \lesssim h_{z_T} \min_{q_z \in \mathcal{P}_1(\mathcal{T}(\hat{\Omega}_{z_T};\mathbb{R}^2))\cap C(\hat{\Omega}_{z_T};\mathbb{R}^2)} \|\nabla(\chi - u_h) - q\|_{L^2(\hat{\Omega}_{z_T})}.$$

The factor $\|J\Lambda_h\|_{L^2(T)}$ can be treated as in (6.14), except in case $u_h = \chi$ on ω_T, which implies $T \in \mathcal{T}_{DC}$. This concludes the proof of Step 3.

Step 4. For any $T \in \mathcal{T}$, it holds

$$\begin{aligned}h_T \|J\Lambda_h\|_{L^2(T)} &\lesssim h_T \|f\|_{L^2(\omega_T)} \\ &\quad + \min_{q_T \in \mathcal{P}_1(\mathcal{T}(\omega_T;\mathbb{R}^2))\cap C(\omega_T;\mathbb{R}^2)} \Big(\|\nabla u_h - q_T\|_{L^2(\omega_T)} + h_T^{1/2} \|(g - q_T \cdot \nu)\|_{L^2(\Gamma_N \cap \partial\omega_T)} \Big), \quad (6.15)\end{aligned}$$

$$\begin{aligned}h_T^2 \|\nabla(J\Lambda_h)\|_{L^2(T)} &\lesssim h_T^2 \|\nabla f\|_{L^2(\omega_T)} \\ &\quad + \min_{q_T \in \mathcal{P}_1(\mathcal{T}(\omega_T;\mathbb{R}^2))\cap C(\omega_T;\mathbb{R}^2)} \Big(\|\nabla u_h - q_T\|_{L^2(\omega_T)} + h_T^{1/2} \|(g - q_T \cdot \nu)\|_{L^2(\Gamma_N \cap \partial\omega_T)} \Big). \quad (6.16)\end{aligned}$$

Proof of Step 4. See Lemma 7 and of Bartels and Carstensen (2004) and observe $J\Lambda_h = \varrho_h$ for their ϱ_h.

Step 5. It holds

$$\begin{aligned}
&\min_{q_T\in\mathcal{P}_1(\mathcal{T}(\omega_T;\mathbb{R}^2))\cap C(\omega_T;\mathbb{R}^2)} \Big(\|\nabla u_h - q_T\|^2_{L^2(\omega_T)} + h_T\,\|g - q_E\cdot\nu\|^2_{L^2(\Gamma_N\cap\partial\omega_T)}\Big)\\
&\quad\lesssim \sum_{E\in\mathcal{E}(\omega_T)} \min_{q_E\in\mathcal{P}_1(\mathcal{T}(\omega_E;\mathbb{R}^2))\cap C(\omega_E;\mathbb{R}^2)} \Big(\|\nabla u_h - q_E\|^2_{L^2(\omega_E)} + h_T\,\|g - q_E\cdot\nu\|^2_{L^2(\Gamma_N\cap E)}\Big)\\
&\quad\lesssim \sum_{E\in\mathcal{E}(\omega_T)} h_E\,\|[\nabla u_h\cdot\nu]\|^2_{L^2(E)} + \sum_{E\in\mathcal{E}(\Gamma_N)} h_E\,\|g - \nabla u_h\cdot\nu\|^2_{L^2(\Gamma_N\cap E)},
\end{aligned}$$

$$\min_{q_z\in\mathcal{P}_1(\mathcal{T}(\widehat{\Omega}_{z_T};\mathbb{R}^2))\cap C(\widehat{\Omega}_{z_T};\mathbb{R}^2)} \|\nabla(\chi - u_h) - q_z\|_{L^2(\widehat{\Omega}_{z_T})} \lesssim \sum_{E\in\mathcal{E}(\Omega_{z_T})} h_E\,\|[\nabla(\chi - u_h)\cdot\nu]\|^2_{L^2(E)}.$$

Proof of Step 5. The first estimate follows from (3.4) in Carstensen and Bartels (2002, p. 951). Consider an interior edge $E\in\mathcal{E}(\Omega)$ and set $q_E := (\nabla u_h|_{T_1} - \nabla u_h|_{T_2})/2$. This yields

$$\|\nabla u_h - q_E\|^2_{L^2(\omega_E)} = 1/4\,\|[\nabla u_h\cdot\nu]\|^2_{L^2(\omega_E)} = |\omega_E|/(4|E|)\;\|[\nabla u_h\cdot\nu]\|^2_{L^2(E)}$$

For any Neumann edge $E\in\mathcal{E}(\Gamma_N)$, ω_E consists of only one element T and we can set $q_E := \nabla u_h|_T$. This proves the second asserted estimate and concludes the proof of Step 5.

Step 6. It holds

$$\begin{aligned}
h_T\,\|f - \Lambda_h\|_{L^2(T)} &\lesssim \|\nabla(w - u_h)\|_{L^2(T)} + \operatorname{osc}(f - \Lambda_h, T) \quad \text{for all } T\in\mathcal{T},\\
h_E^{1/2}\,\|[\nabla u_h\cdot\nu]\|_{L^2(E)} &\lesssim \|\nabla(w - u_h)\|_{L^2(\omega_E)} + \operatorname{osc}(f - \Lambda_h, \mathcal{T}(E)) \quad \text{for all } E\in\mathcal{E}(\Omega),\\
h_E^{1/2}\,\|\nabla u_h\cdot\nu - g\|_{L^2(E)} &\lesssim \|\nabla(w - u_h)\|_{L^2(\omega_E)} + \operatorname{osc}(f - \Lambda_h, \mathcal{T}(E)) + \operatorname{osc}(g, E)\\
&\qquad\qquad \text{for all } E\in\mathcal{E}(\Gamma_N).
\end{aligned}$$

Proof of Step 6. Those estimates follow from an error analysis for the residual-based error estimator with bubble functions as in Subsection 3.4.2.

End of proof of Theorem 6.3.6. The combination of Step 1-6, overlap arguments and Cauchy inequalities conclude the proof of Theorem 6.3.6. □

This section concludes with some remarks to discuss some quantities in Theorem 6.3.6 that appear critical. In model examples, such as the benchmarks with affine obstacles in Section 6.4, they are in fact noncritical and do not spoil the equivalence of GUB and $|||u - u_h||| + \|\varrho - \varrho_h^\star\|_\star$ up to higher-order terms.

Remark 6.3.7 (Remarks on the critical terms of Theorem 6.3.6)**.** *(a) Since the Riesz representation Λ of ϱ from* (6.4) *is merely an $L^2(\Omega)$ function, it is not clear a priori that the term $\|\varrho\circ(\mathbb{I} - J)\|_\star$ is of higher order. Of course, in the case $\Lambda\in L^2(\Omega)$, Lemma 6.3.2.(c) yields*

$$\|\varrho\circ(\mathbb{I} - J)\|_\star = \sup_{v\in V} \varrho(v - Jv)/\,|||v||| \leqslant \operatorname{osc}(\Lambda, \mathcal{N}).$$

For the general case, consider the sets $\mathcal{C}$, $\mathcal{D}$ and the free boundary $\mathcal{F} := \Omega\setminus(\mathcal{C}\cup\mathcal{D})$ from (6.3)

and observe

$$\varrho(v - Jv) = \sum_{z\in\mathcal{N}} \varrho((v - v_z)\varphi_z).$$

The contributions for $z \in \mathcal{N}$ with $\omega_z \subset (\mathcal{C} \cup \mathcal{D})$ allows for the local Riesz representation from (6.4) *and hence*

$$\sum_{\substack{z\in\mathcal{N}\\ \omega_z\subset(\mathcal{C}\cup\mathcal{D})}} \varrho((v - v_z)\varphi_z) \leqslant \operatorname{osc}(f,\mathcal{N})\,|||v|||\,.$$

So it remains to analyse the contributions from nodes in

$$\mathcal{N}(\mathcal{F}) \coloneqq \{z \in \mathcal{N} \,\big|\, \omega_z \cap \mathcal{F} \neq \varnothing\}.$$

In simple model scenarios, $\mathcal{F}$ is indeed a one-dimensional submanifold, cf. Examples 1-2 in Section 6.4, where ϱ is piecewise smooth and bounded. Therefore, we expect

$$\varrho((v - v_z)\varphi_z) \approx |\omega_z|\,\|\varphi_z\nabla v\|_{L^2(\omega_z)}\,.$$

For the local mesh size $h(s)$ along the parameterisation of the curve $\mathcal{F}$ by arc-length $0 \leqslant s \leqslant L \coloneqq |\gamma| \lesssim 1$, it holds $|\omega_z| \approx h(s)^2$ near any point $\gamma(s) \in \omega_z$. The node patches along $\mathcal{F}$ are coupled with $J \coloneqq |\mathcal{N}(\mathcal{F})| + 1$ points $\gamma(t_0), \dots, \gamma(t_J)$, along γ with $0 = t_0 < t_1 < \dots < t_J = L$. The nonsmooth contributions in the neighbourhood of γ sum up to

$$\begin{aligned}
\sum_{z\in\mathcal{N}(\mathcal{F})} \varrho((v - v_z)\varphi_z) &\leqslant \left(\sum_{z\in\mathcal{N}(\mathcal{F})} h_z^4\right)^{1/2} |||v||| \lesssim \left(\sum_{j=0}^{J} h(t_j)^4\right) |||v||| \\
&\lesssim \left(\sum_{j=0}^{J} h(t_j)^3 (t_{j+1} - t_j)\right) |||v||| \quad (\text{with } t_{J+1} \coloneqq L + h(L)) \\
&= \int_0^L h(s)^3 \,\mathrm{d}s\, |||v||| \leqslant L\, h_{max}^3\, |||v||| \lesssim h_{max}^3\, |||v|||\,.
\end{aligned}$$

Here h_{max} denotes the maximal mesh size along γ, which is relatively small compared with the maximal mesh size $\max_{T\in\mathcal{T}} h_T$ of the triangulation $\mathcal{T}$ for all the adaptive meshes in the numerical examples of Section 6.4. Therefore,

$$|||\varrho \circ (\mathbb{I} - J)|||_\star \lesssim h_{max}^{3/2}$$

is of higher order compared to $|||u - u_h|||$.
(b) The contribution $\|\nabla w_D\|_{L^2(T)}\,\|h_{\mathcal{T}} f\|_{L^2(T)}$ arises only for a relatively small number of triangles along the Dirichlet boundary. It vanishes for boundary triangles $T \notin \mathcal{T}_{DC}$ without contact at the Dirichlet boundary. It also vanishes for piecewise affine Dirichlet data u_D.
(c) The contribution $\|\nabla(u - \chi)\|_{L^2(\mathcal{T}_N)}$ vanishes for pure Dirichlet boundary problems with $\Gamma_N = \varnothing$. This is valid for all benchmark examples in Section 6.4. It also vanishes for triangles without contact at the Neumann boundary in its neighbourhood ω_T.

6.4 Numerical Examples

This section reports on results in five numerical benchmark examples. The adaptive mesh refinement is driven by the Dörfler marking of Subsubsection 2.3.4.2 with the elementwise refinement indicators

$$\begin{aligned}\eta(T)^2 \coloneqq \frac{|T|}{\lambda_{\max,T}}\|f-\Lambda_h\|^2_{L^2(T)} + |T|^{1/2}\sum_{E\in\mathcal{E}(T)}\lambda^{-1}_{\max,T}\|[\sigma_h\cdot\nu_E]\|^2_{L^2(E)}\\ + \int_T(\chi-u_h)\min(J\Lambda_h,0)\,\mathrm{d}x + \sum_{z\in\mathcal{N}(T)}\|\varphi_z(\Lambda_h-(\Lambda_h)_{\omega_z})\|_{L^2(\omega_z)}/3 \qquad (6.17)\\ + 0.248\left\|h_{\mathcal{E}}^{3/2}\partial^2 u_D/\partial s^2\right\|^2_{L^2(\partial T\cap\Gamma_D)}.\end{aligned}$$

The application of the error estimators η_{xyz} of Chapter 3 for the estimation of $\mathrm{Res}_{\mathrm{aux}}$ from Theorem 6.3.3 leads to the guaranteed upper bounds

$$\mathrm{GUB}(\eta_{\mathrm{xyz}}) \coloneqq \alpha/2+\sqrt{\alpha^2/4+\beta}+0.498\left\|h_{\mathcal{E}}^{3/2}\partial^2 u_D/\partial s^2\right\|_{L^2(\Gamma_D)} \qquad (6.18)$$

with

$$\begin{aligned}\alpha &\coloneqq \eta_{\mathrm{xyz}}+\mathrm{osc}(\Lambda_h,\mathcal{N})+0.498\left\|h_{\mathcal{E}}^{3/2}\partial^2 u_D/\partial s^2\right\|_{L^2(\Gamma_D)},\\ \beta &\coloneqq \int_T(\chi-u_h)\min(J\Lambda_h,0)\,\mathrm{d}x.\end{aligned}$$

Note that η_{xyz} might include additional overhead terms with $f-\Lambda_h$.

6.4.1 Square Domain

The first benchmark from Nochetto et al. (2003) concerns the constant obstacle $\chi=\mathcal{I}\chi\equiv 0$ on the square domain $\Omega=(-1,1)^2$ subject to smooth Dirichlet data $u_D(r,\varphi)=r^2-0.49$ and right-hand side

$$f(r,\varphi)=\begin{cases}-16r^2+3.92 & \text{for } r>0.7\\ -5.8408+3.92r^2 & \text{for } r\leqslant 0.7.\end{cases}$$

The exact solution reads

$$u(r,\varphi)=\max\{0,r^2-0.49\}^2$$

and touches the obstacle outside the circle with radius 0.7.

Figure 6.1 displays the convergence history of the exact error for uniform and adaptive mesh refinement as well as the three overhead terms with respect to the number of degrees of freedom $|\mathcal{M}|$. Since the solution is smooth, the energy error converges with the optimal empirical convergence rate 1/2 also for uniform mesh refinement and there is only minor improvement by adaptive mesh refinement. All overhead terms are of higher order with

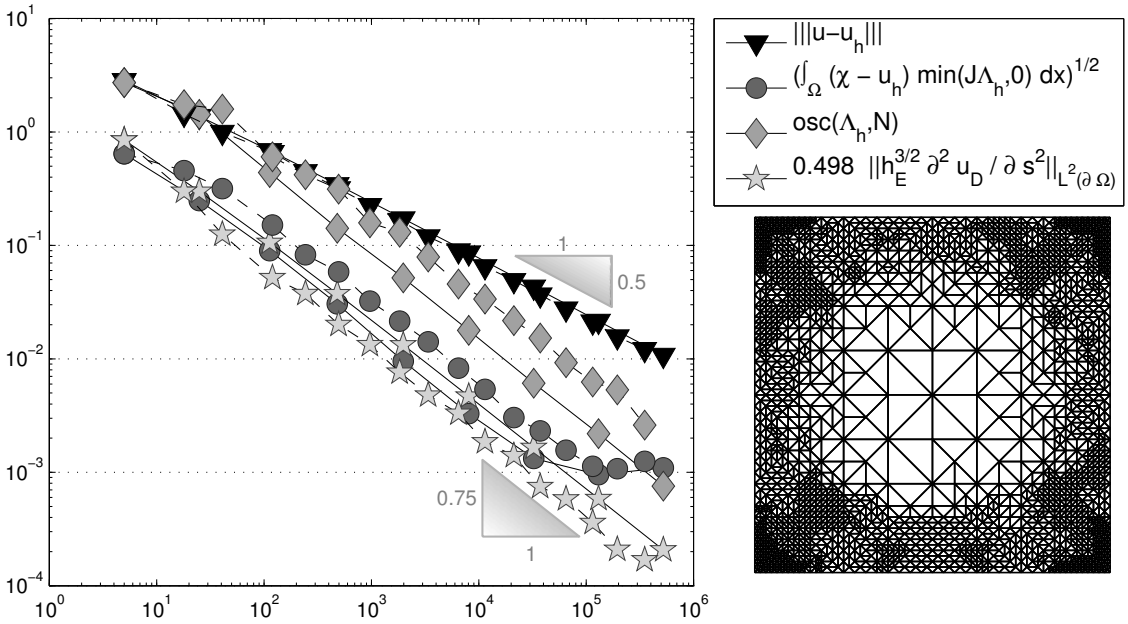

Figure 6.1: Convergence history for the energy error $|||u - u_h|||$ and some overhead terms on uniform (solid line) and adaptive (dashed line) meshes in Subsection 6.4.1 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 7.

convergence rate 3/4. However, the overhead term $\beta^{1/2}$ stagnates after around 10^5 degrees of freedom. Since $\beta^{1/2}$ is very small compared to $|||u - u_h|||$ this is not a problem and undocumented experiments convey that this is related to inexact solve. One adaptive mesh is displayed in Figure 6.1 on the right-hand side. The contact zone $\{r < 0.7\}$ is less refined which appears reasonable, because there is no error within the contact zone. Since there is no contact along the boundary $\partial\Omega$, the critical boundary term of Remark 6.3.7.(b) does not arise.

Figures 6.2 and 6.3 compare the efficiency indices $\mathrm{GUB}(\eta_{\mathrm{xzy}})/|||e|||$ of the global upper bounds $\mathrm{GUB}(\eta_{\mathrm{xzy}})$ from (6.18) The efficiency indices are around 3.5 but decrease slowly to values between 1 and 1.5. This is due to the decrease of the extra terms and consistent with the observation that the relative contribution of η_{xyz} becomes more and more dominant. As a consequence, there is a significant impact of the accuracy of η_{xyz} on the efficiency of the global upper bound $\mathrm{GUB}(\eta_{\mathrm{xzy}})$.

6.4.2 L-Shaped Domain

The second benchmark example from Bartels and Carstensen (2004) mimics a typical corner singularity on the L-shaped domain $\Omega = (-2,2)^2\backslash([0,2] \times [-2,0])$ with constant obstacle $\chi = \chi_h \equiv 0$ and homogeneous Dirichlet data $u_D \equiv 0$ along $\partial\Omega$, with the right-hand side

$$f(r,\varphi) \coloneqq -r^{2/3}\sin(2\varphi/3)\left(7(\partial g(r)/\partial r)/(3r) + (\partial^2 g(r)/\partial r^2)\right) - H(r-5/4),$$
$$g(r) \coloneqq \max\{0, \min\{1, -6s^5 + 15s^4 - 10s^3 + 1\}\}$$

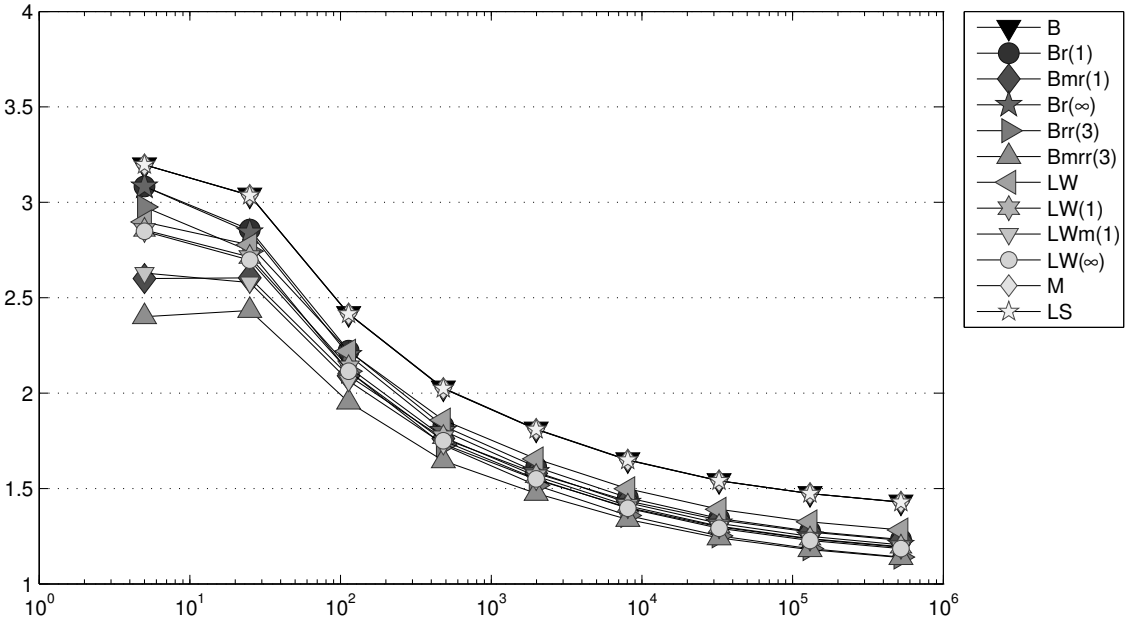

Figure 6.2: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 6.4.1.

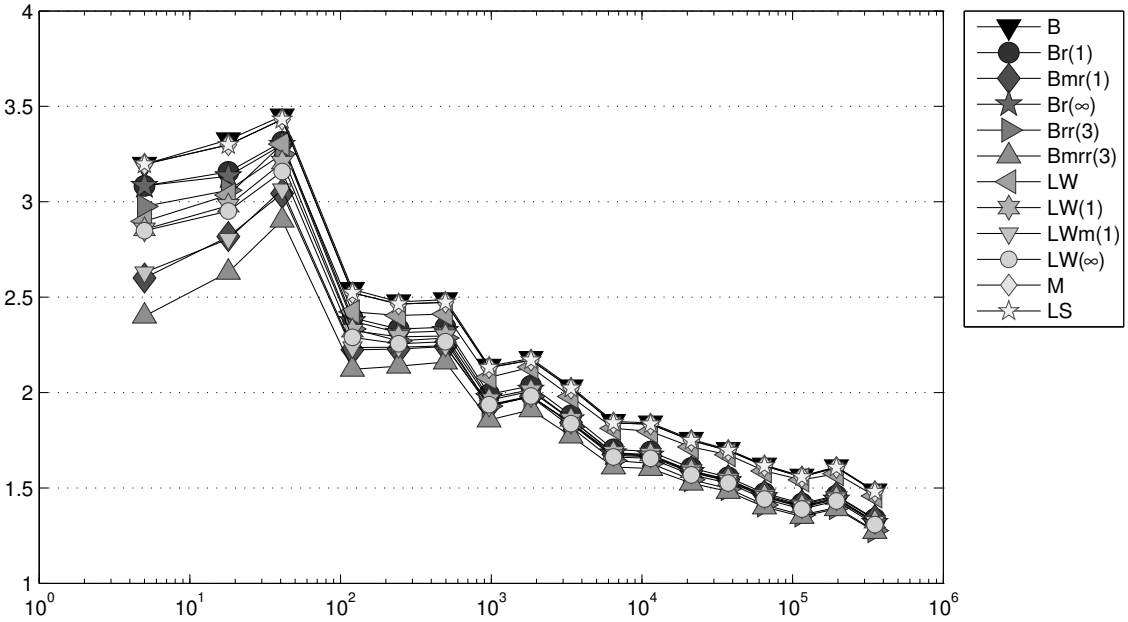

Figure 6.3: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 6.4.1.

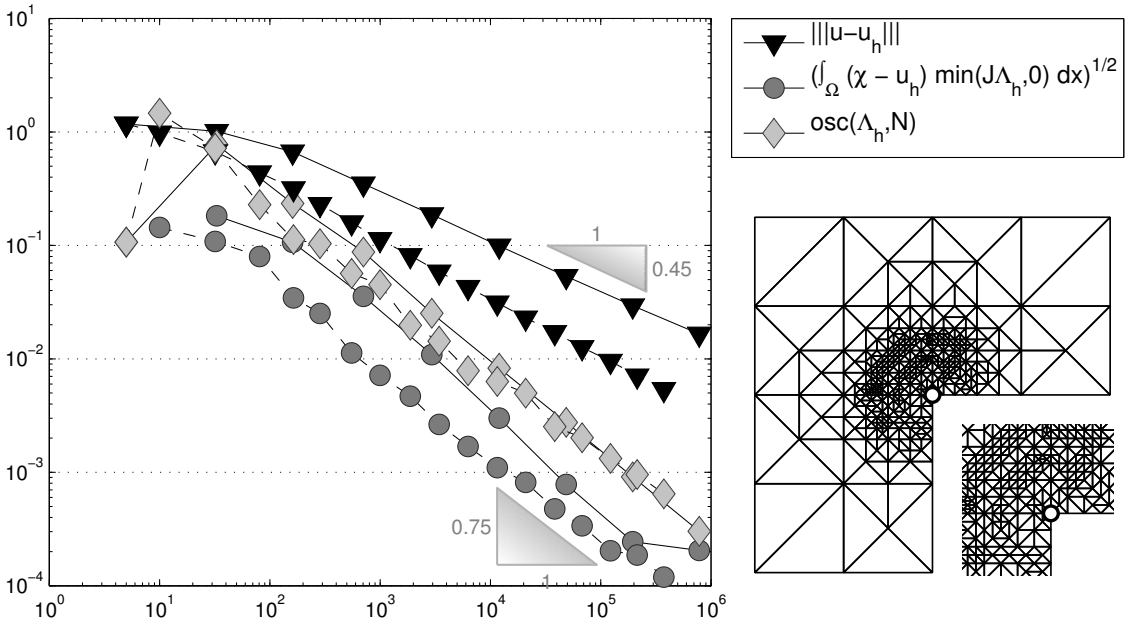

Figure 6.4: Convergence history for the energy error $|||u - u_h|||$ and some overhead terms on uniform (solid line) and adaptive (dashed line) meshes in Subsection 6.4.2 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 5 and the neighbourhood of the singular point $(0, 0)$ magnified by a factor 2.

for $s := 2(r - 1/4)$ and the Heaviside function H. The exact solution reads

$$u(r, \varphi) := r^{2/3} g(r) \sin(2\varphi/3).$$

The contact zone $\{r > 3/4\}$ has a nonvoid intersection with the boundary $\partial\Omega$. The critical boundary term of Remark 6.3.7.(b) does not arise due to $w_D \equiv 0$.

Figure 6.4 conveys that the experimental convergence rate of the energy error for uniform mesh refinement is about 0.4. Adaptive mesh refinement improves it to the optimal value 1/2 and shortens the pre-asymptotic range. The overhead terms are of higher order.

The efficiency indices depicted in Figures 6.5 and 6.6 range from 1 to 4. The best error estimator is $\eta_{\mathrm{Bmrr}(3)}$ which assumes efficiency indices around 1.1 for fine meshes, but also yields much better indices for coarse meshes due to the mean correction and the postprocessing.

6.4.3 Cusp Obstacle on Square Domain

The third example taken from Nochetto et al. (2003) involves Ω, f and u_D from Subsection 6.4.1 and the cusp-shaped obstacle

$$\chi := \max\{-2, 1 - 50 \max\{|x|, |y|\}\}.$$

Since the exact solution is unknown, it is replaced by the solution on the two times red-refined triangulation $\mathrm{red}^2(\mathcal{T})$ for the computation of the energy error $|||u - u_h|||$ on $\mathcal{T}$. The

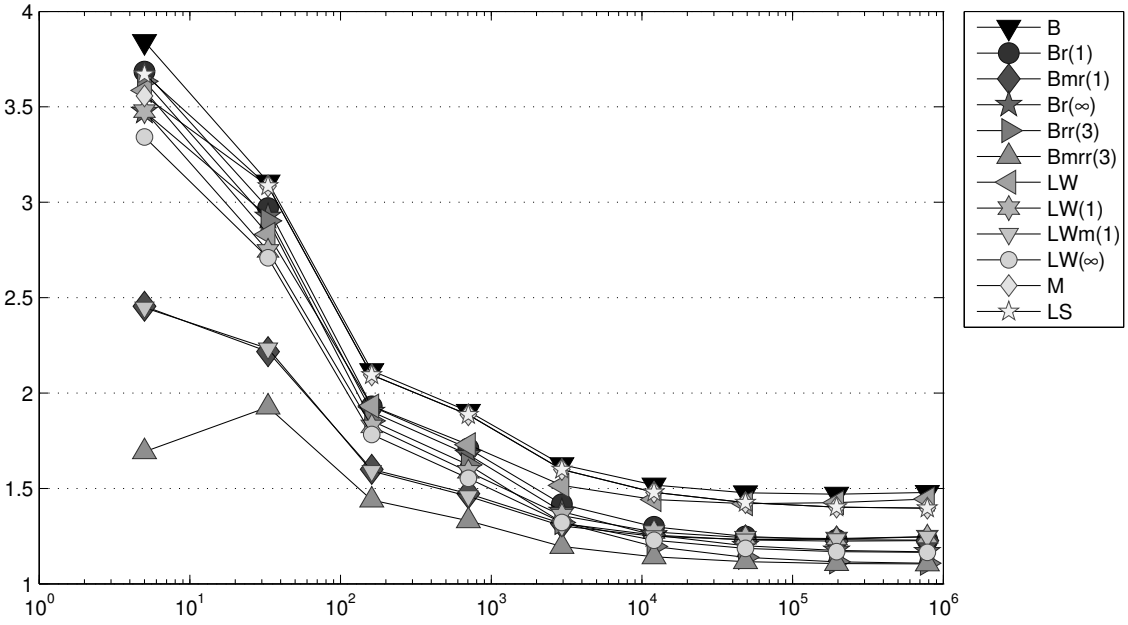

Figure 6.5: History of efficiency indices $\mathrm{GUB}(\eta_{\mathrm{xyz}})/|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 6.4.2.

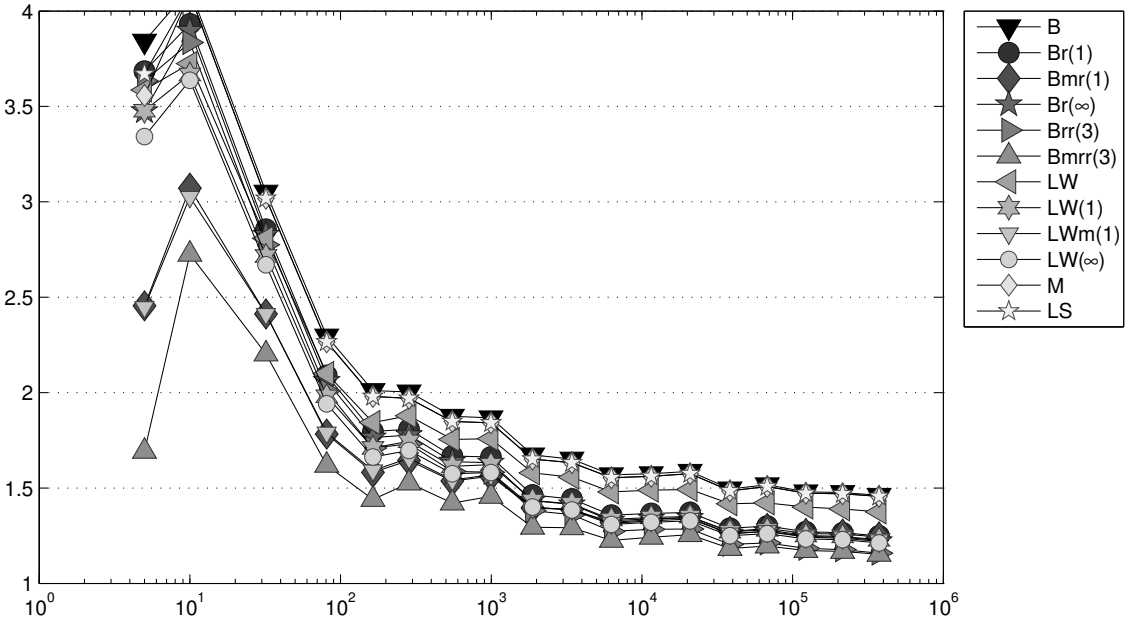

Figure 6.6: History of efficiency indices $\mathrm{GUB}(\eta_{\mathrm{xyz}})/|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 6.4.2.

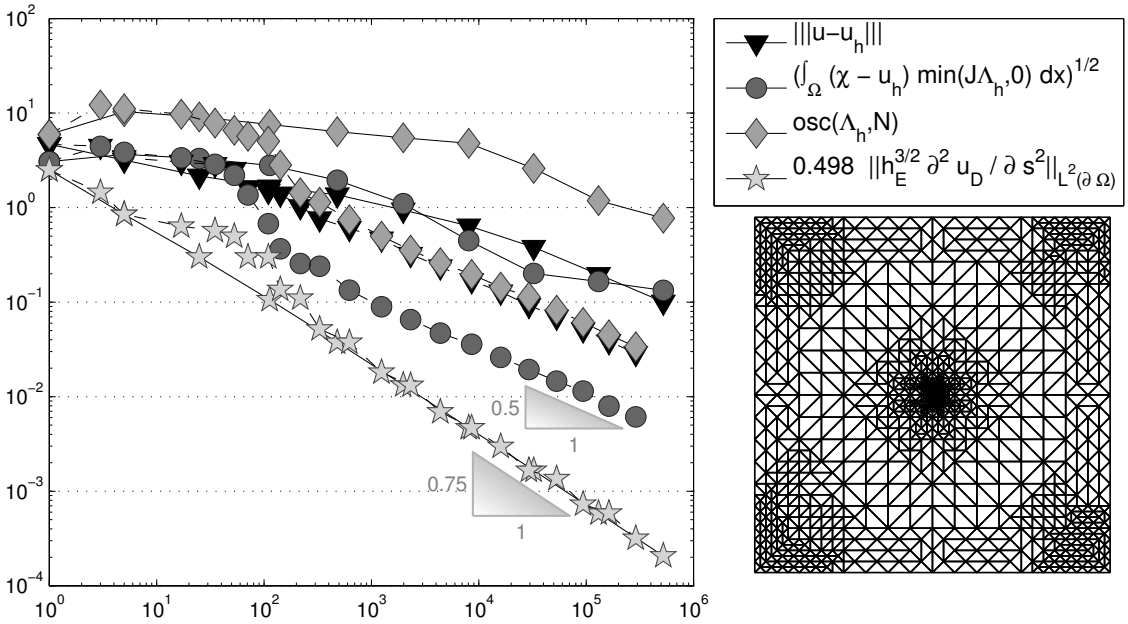

Figure 6.7: Convergence history for the energy error $|||u - u_h|||$ and some overhead terms on uniform (solid line) and adaptive (dashed line) meshes in Subsection 6.4.3 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 12.

obstacle χ is piecewise affine, but not on the coarse initial triangulation. However, $\chi \leqslant \mathcal{I}\chi$ leads to a conforming discretisation and reliable guaranteed upper bounds $\mathrm{GUB}(\eta_{\mathrm{xyz}})$.

Figure 6.7 indicates that the adaptive mesh refinement algorithm recovers the optimal empirical convergence rate of the energy error and shortens the pre-asymptotic range. In this example, the overhead term $\mathrm{osc}(\Lambda_h, \mathcal{N})$ is not of higher order, which is a strong indication that the heuristic argument of Remark 6.3.7.(a) fails for nonsmooth obstacles as in this example. Figure 6.8 shows that the efficiency indices of all global upper bounds increase beyond 10 with the number of degrees of freedom in case of uniform mesh refinement. For adaptive mesh refinement the efficiency indices stay around 2.5 after a strong decrease from 6 at about 100 unknowns in Figure 6.9. This fall is connected to the sudden decrease of the extra terms, especially $\mathrm{osc}(\Lambda_h, \mathcal{N})$, which might be caused by the gradual revelation of the real obstacle χ by the adaptive mesh refinement.

6.4.4 Pyramid Obstacle on Square Domain

In order to explore the limitations of the theoretical results, the fourth benchmark employs the constant right-hand side $f \equiv 1$ and the nonaffine obstacle $\chi(x, y) = \mathrm{dist}((x, y), \partial\Omega)$ from Bartels and Carstensen (2004) on the square domain $\Omega = (-1, 1)^2$ with homogeneous Dirichlet data $u_D \equiv w_D \equiv 0$ on $\Gamma_D := \partial\Omega$. The initial triangulation consists of four elements such that $\chi = \chi_h$. Since the exact solution is unknown, it is replaced by the solution on the two times red-refined triangulation $\mathrm{red}^2(\mathcal{T})$ for the computation of the energy error $|||u - u_h|||$ on $\mathcal{T}$.

In contrast to the first two benchmarks, the obstacle is not globally affine and the contact

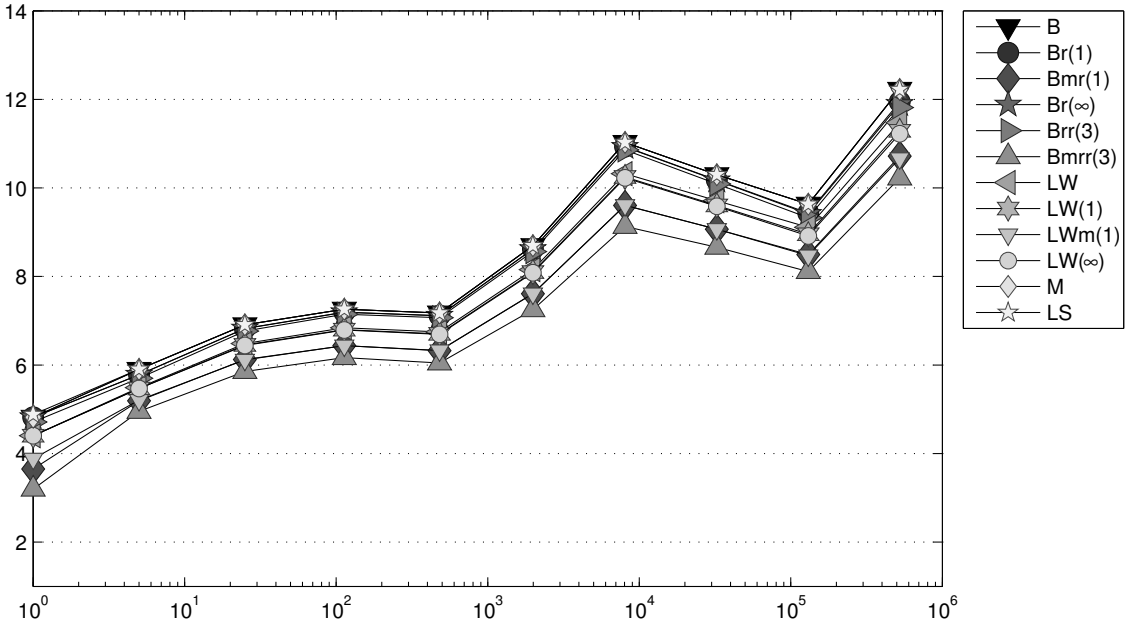

Figure 6.8: History of efficiency indices $\mathrm{GUB}(\eta_{\mathrm{xyz}})/|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 6.4.3.

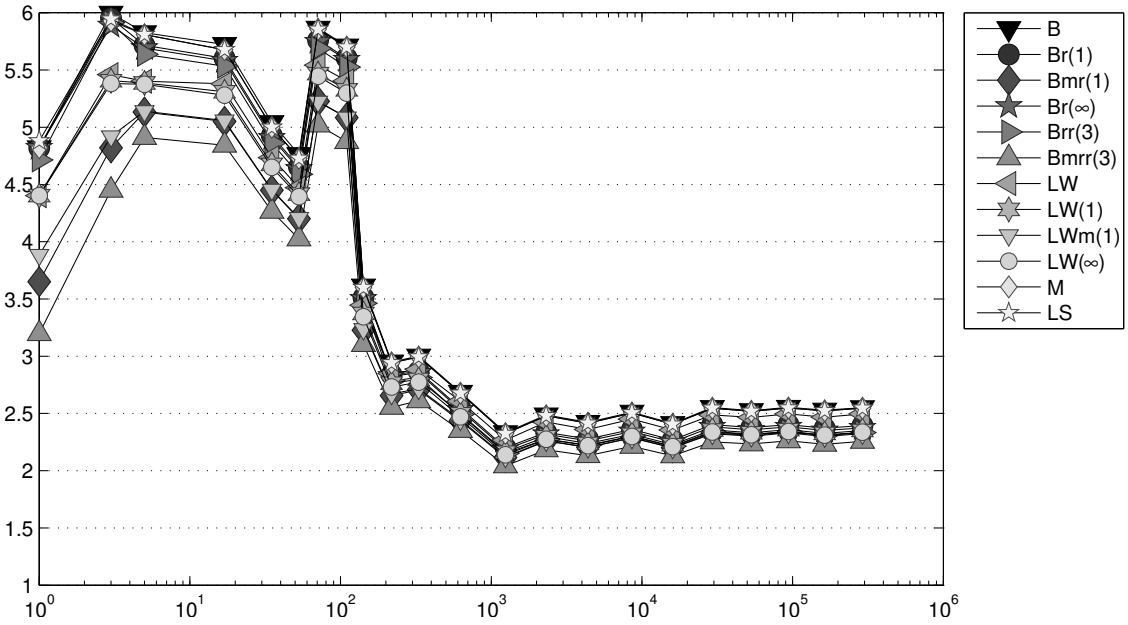

Figure 6.9: History of efficiency indices $\mathrm{GUB}(\eta_{\mathrm{xyz}})/|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 6.4.3.

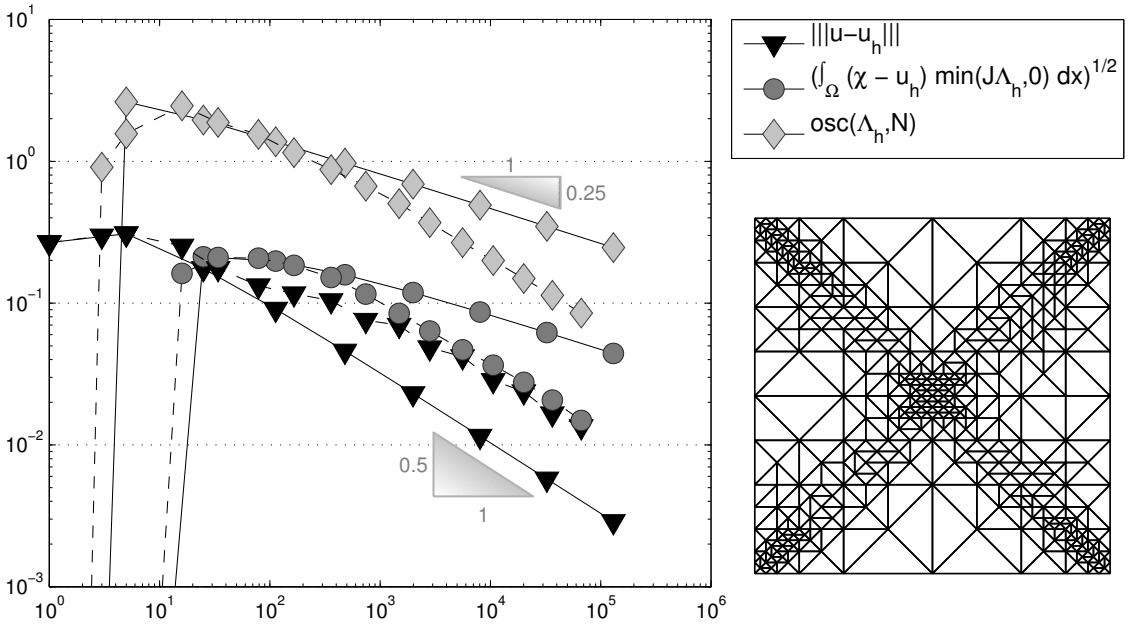

Figure 6.10: Convergence history for the energy error $|||u - u_h|||$ and some overhead terms on uniform (solid line) and adaptive (dashed line) meshes in Subsection 6.4.4 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 7.

zone reduces to the lines

$$\{(x, y) \in (0, 1)^2 \,\big|\, y = x \text{ or } y = 1 - x\}.$$

While uniform mesh refinement yields the optimal empirical convergence rate, the adaptive process has a rather long stagnating pre-asymptotic range as shown in Figure 6.10! The adaptive mesh in Figure 6.10 shows a strong refinement along the contact edges due to very high contributions of the extra terms and nonvanishing edge jumps of ∇u on these edges. A similar behaviour was observed by Bartels and Carstensen (2004) and is expected for every error estimator that is based on edge jumps of ∇u_h.

Since the error estimators have been derived for affine obstacles, the efficiency result of Section 6.3.3 cannot be expected to hold. In fact, Figure 6.11 indicates that the upper bound $\mathrm{GUB}(\eta_{\mathrm{xyz}})$ is not efficient with respect to $|||e|||$. The efficiency indices blow up (over 100) for uniform mesh refinement. Adaptive mesh refinement seems to restore the efficiency with efficiency indices around 10 as shown in Figure 6.12, but this is still not rewarding regarding the poor results on the actual error reduction on the produced meshes.

6.4.5 Nonaffine Smooth Obstacle

The last benchmark illustrates that the global upper bound is also applicable to problems with smooth obstacles. The example from Gräser and Kornhuber (2009) considers the obstacle $\chi(x, y) = -(x^2 - 1)(y^2 - 1)$ and $f \equiv -\Delta\chi$ on the square domain $\Omega = (-1, 1)^2$ with the exact solution $u \equiv \chi$.

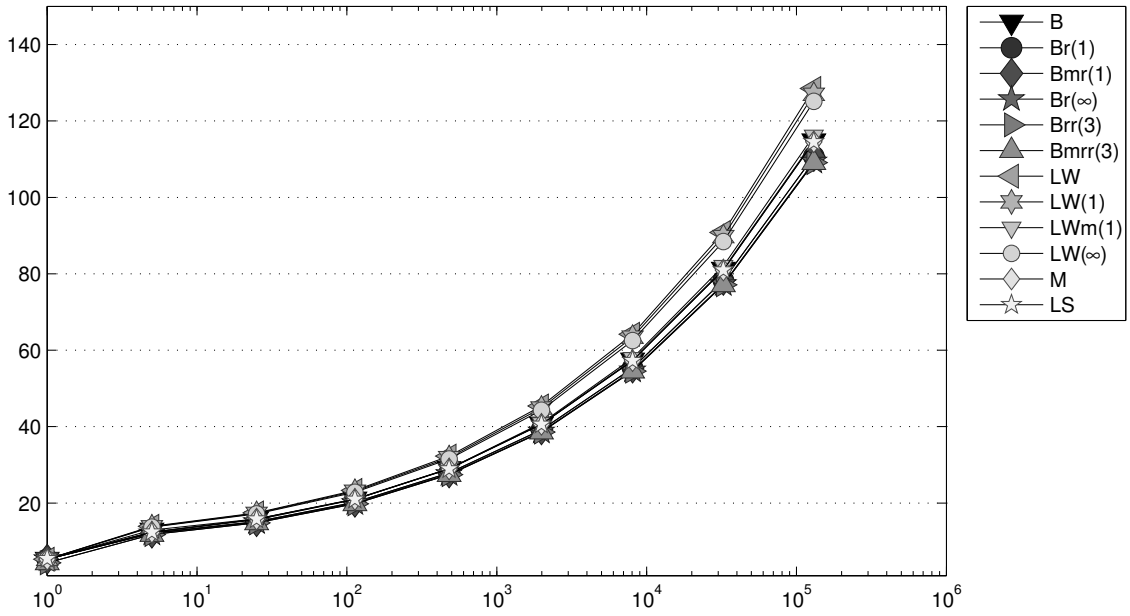

Figure 6.11: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 6.4.4.

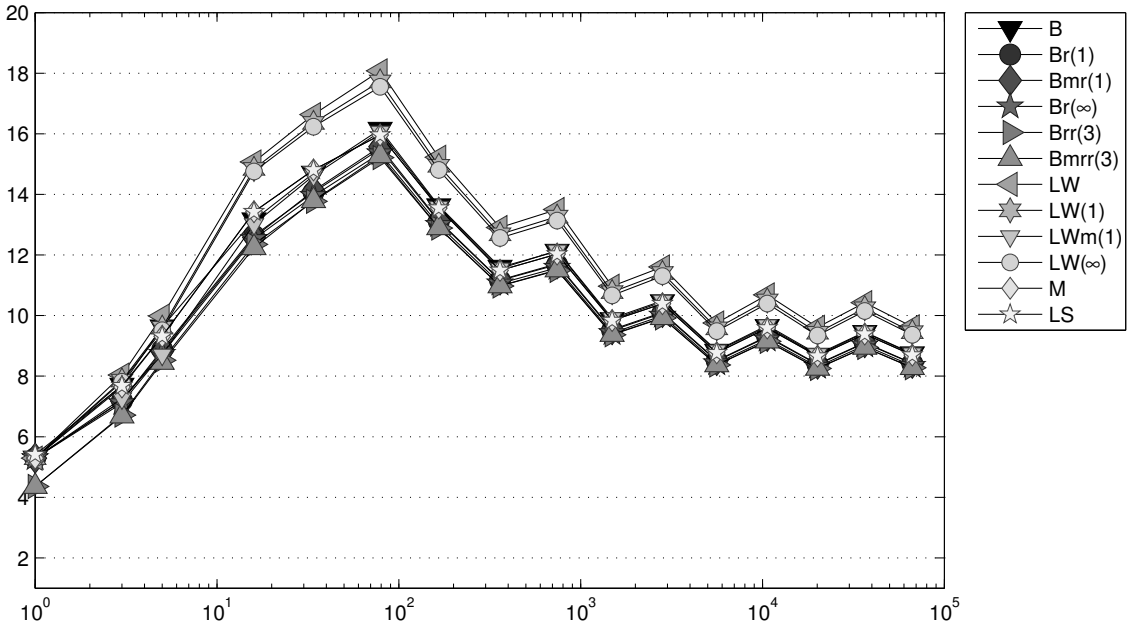

Figure 6.12: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 6.4.4.

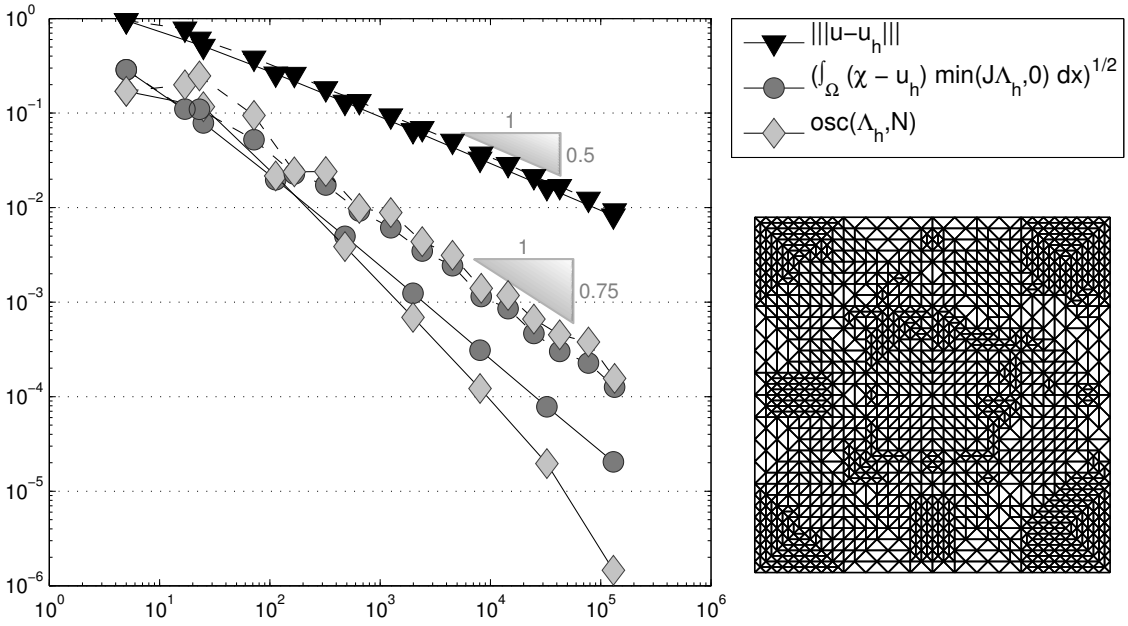

Figure 6.13: Convergence history for the energy error $|||u - u_h|||$ on uniform (solid line) and adaptive (dashed line) meshes in Subsection 6.4.5 with respect to the number of degrees of freedom $|\mathcal{M}|$. The right image shows the adaptive mesh on level 7.

Due to the homogeneous Dirichlet boundary conditions, w_D could be set to zero. But this is a nonconforming obstacle problem with possibly $u_h \notin \mathcal{K}$. Hence, the reliability would not hold. Instead, the choice $w_D := -\min\{0, u_h - \chi\} \geqslant 0$ in Theorem 6.3.3 after Falk (1974) and Remark 6.3.4.(e) leads to an admissible test function $u_h + w_D \in \mathcal{K}$ and the same global upper bound with the extra terms (only for this subsection)

$$b := \left(\int_\Omega (\chi - u_h - w_D) J\Lambda_h \, \mathrm{d}x \right)^{1/2} \quad \text{and} \quad |||w_D||| := |||\min\{0, u_h - \chi\}||| \,.$$

Figure 6.13 shows that the adaptive mesh refinement barely worsens the empirical convergence rate. Figures 6.14 and 6.15 display that the efficiency indices are not as good as in the affine examples due to the contribution $|||w_D|||$, which is not of higher order compared to $|||u - u_h|||$. But there is a significant improvement of the efficiency indices through adaptive mesh refinement, which reduces the relative contribution $|||w_D|||$ to the global upper bound.

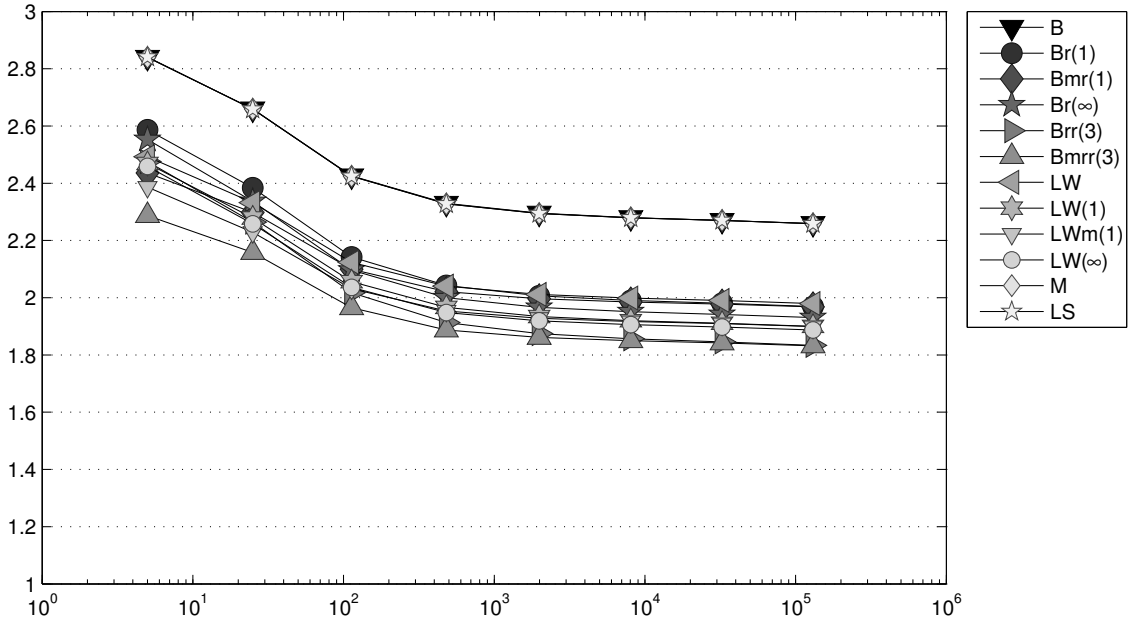

Figure 6.14: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on uniform meshes in Subsection 6.4.5.

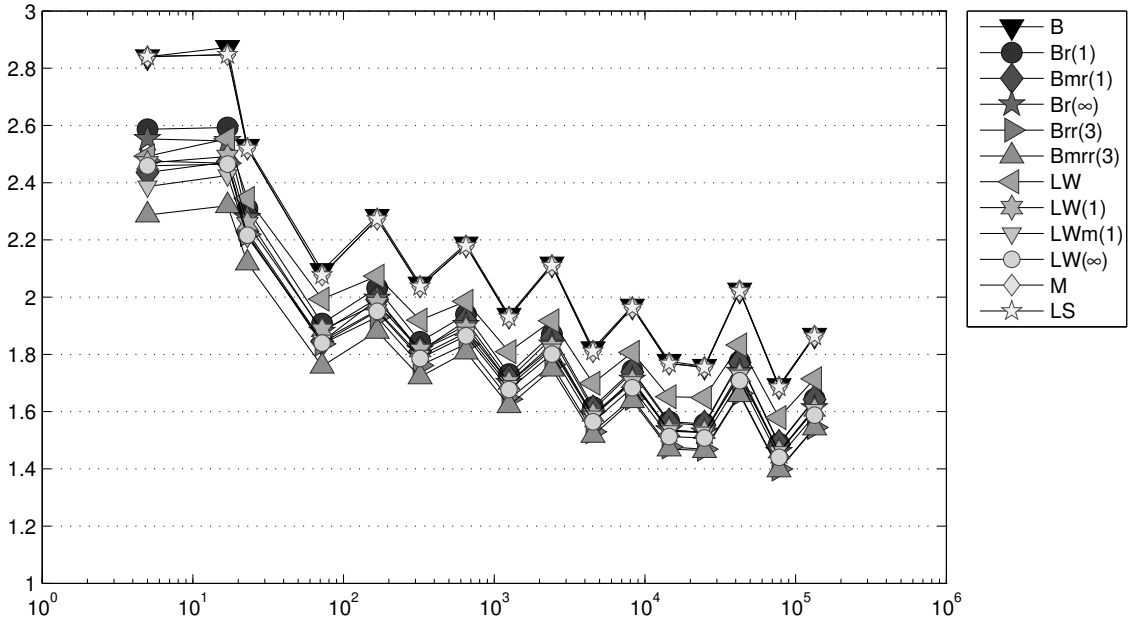

Figure 6.15: History of efficiency indices GUB(η_{xyz})/ $|||e|||$ of various error estimators η_{xyz} labelled *xyz* as functions of the number of degrees of freedom on adaptive meshes in Subsection 6.4.5.

A MATLAB Implementation

This chapter contains the main parts of the MATLAB code and the additions to AFEM that were made for this thesis. The first section describes the requirements to set up, solve and estimate a problem given by the user. The remaining sections explain essential parts of the implementation. To save some space, some code lines in the displayed Listings below were combined or differently printed and also the comments are not displayed here. The complete code can be found on the attached Compact Disc[1]. All numerical experiments of this thesis have been conducted with MATLAB (Version Version 7.10.0.499 (R2010a)).

A.1 Setup of a Problem in AFEM

This section explains how to use the AFEM version of this thesis to solve a Poisson, Stokes or obstacle problem.

Table A.1 shows the list of all available solvers and the required input data additional to the data of the initial triangulation of the domain Ω. The structure of the triangulation data sets is explained in the AFEM introduction in Subsection 2.3.5. The remaining data is expressed via function handles. Note that by now the diffusion tensor S was implemented only as a scalar-valued function handle called `alpha`. However, the theory in the thesis is valid for matrix-valued diffusion tensors and an extension to matrix-valued function handles may be done in the future.

Listing A.1 shows a simple example that loads the geometry data for an L-shaped domain in Line 1 and defines function handles for $f \equiv 1$, $g \equiv 0$ and $u_D \equiv 0$ as well as $\alpha \equiv 1$. Line 6 calls the solver for the conforming $\mathcal{P}_1$ finite element method for the Poisson problem. The output `x` is a vector with the nodal values of the discrete solution with respect to the triangulation given by `c4n` and `n4e`. Line 7 plots the discrete solution with the AFEM function `plotP1`. Lines 8 and 9 solve and plot the discrete solution of the Crouzeix-Raviart nonconforming FEM. Other possible solvers and the required data are listed in Table A.1. The MATLAB code for the solvers for the Stokes problem and the obstacle problem are discussed in Chapter 5 and Chapter 6, respectively.

Table A.2 lists all benchmark problems of this thesis and the filenames wherin the corresponding function handles for the data and the exact solution (if available) are specified. They also call the AFEM loop for the error estimator competition.

[1] The online version of this document contains the content of the Compact Disc as embedded zip-file. Please use an appropriate pdf viewer to extract the file, such as KDE Okular or Adobe Reader. Then, rename the file to `code.zip` and unpack the file with adequate Software, such as WinZip. In contrast to the thesis' text, the code is provided under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. Refer to the file `LICENSE.txt` on the the Compact Disc or in the zip-file for more information.

```
[c4n n4e n4sDb n4sNb]=loadGeometry('LshapeNb',3);
f=@(x)(sin(pi*x(:,1)).*cos(pi*x(:,2))); degree_f = 10;
g=@(x,normals)(sin(pi*x(:,1)).*cos(pi*x(:,2))); degree_g = 10;
alpha=@(x)(ones(length(x),1));
u4Db=@(x)(zeros(length(x),1)); degree_u4Db = 0;
TD2u4Db=@(x,tangents)(zeros(length(x),1));
[x,~,~,~,sigma4e]=solveP1Poisson(f,g,u4Db,alpha,c4n,n4e,n4sDb,n4sNb,degree_f,degree_g,);
plotP1(c4n,n4e,x);
[x,~,~,~,sigma4e]=solveCRPoisson(f,g,u4Db,alpha,c4n,n4e,n4sDb,n4sNb,degree_f,degree_g,degree_u4Db);
plotCR(c4n,n4e,x);
```

Listing A.1: Example for user-defined problem.

Filename	Description	Data
`solveP1Poisson`	$\mathcal{P}_1$-FEM for Poisson problem	`f,g,u4Db,alpha`
`solveCRPoisson`	CR-NCFEM for Poisson problem	`f,g,u4Db,alpha`
`solveMINIStokes`	Mini-FEM for Stokes problem	`f,u4Db`
`solveCRPOStokes`	CR-NCFEM for Stokes problem	`f,u4Db`
`solveP1Obstacle`	$\mathcal{P}_1$-FEM for Obstacle problem	`f,g,u4Db`

Table A.1: List of available solvers in the `AFEM` package and the data function handles. The function handle `f` denotes the right-hand side source function, `g` denotes Neumann boundary data, `u4Db` denotes Dirichlet boundary data and `alpha` denotes the scalar diffusion parameter. Additional optional degrees for the integration are not displayed and set to zero if not specified.

A.2 General Remarks on Error Estimators

Now that the discrete solution is known, the next step is error estimation. For convenience, the solvers also return the piecewise constant stress tensor σ_h in form of an array `sigma4e` of the dimension $|\mathcal{T}| \times 2$. Most error estimators need `sigma4e` (as well as the data function handles) as an input parameter to compute the error estimator. Table A.3 lists all implemented error estimators, their filenames and the references for the theoretical background above in this thesis. Note that these functions in general do not compute additional overhead terms like oscillations $\operatorname{osc}(f,\mathcal{T})$ of f or Dirichlet boundary error contributions. The overhead quantities are computed in extra functions that are discussed in Section A.10. As an example of how to call an error estimator with all overhead terms,

Listing A.2 displays the `MATLAB` code to obtain the Braess error estimator η_B from Definition 3.2.7 for a Poisson problem. Line 3 calls a function that computes the main part of the error estimator, i.e. $\left\|\mathbb{S}^{-1/2}(\sigma_h - q_B)\right\|_{L^2(\Omega)}$. The function is explained in Section A.3. Lines 4–17 add the overhead terms with the Poincaré constant `CP`=$1/j_{1,1} = 0.2610$ for the volume oscillations and `CNb`=1.1473 from Table 3.2 for the oscillations of the Neumann data. Lines 18–25 add the Dirichlet error contribution from Theorem 4.2.1 with the constant `CDb`=0.4980 from Theorem 4.2.2. The function handles `f`,`g`, `u4Db`,`alpha` and `TD2u4Db` for the second tangential derivative of u_D are assumed to be defined beforehand as in Listing A.1. The Compact Disc contains an exemplary `MATLAB` script `afemExample.m` that extends Listing A.2 to a comparison between the Braess error estimator without postprocessing and the postprocessed Braess error estimator inside an

Filename	Features	Subsection
Poisson problems for $\mathcal{P}_1$ conforming finite element method		
`afemPoisson_Lshape`	Reentrant corner	4.3.1
`afemPoisson_SquareOSC`	Large oscillations	4.3.2
`afemPoisson_SquareJumps`	jumps on square	4.3.3
`afemPoisson_OktagonJumps`	jumps on octagon	4.3.4
`afemPoisson_Sector`	curved boundaries	4.6.1
Poisson problems for Crouzeix-Raviart nonconforming finite element method		
`afemPoissonCR_Lshape`	Reentrant corner	4.5.2
`afemPoissonCR_SquareOSC`	Large oscillations	4.5.3
`afemPoissonCR_SquareJumps`	Jumps on square	4.5.4
`afemPoissonCR_OktagonJumps`	Jumps on octagon	4.5.5
`afemPoissonCR_Sector`	Curved boundaries	4.6.2
Stokes problems for mini finite element method		
`afemStokesMINI_Lshape`	Reentrant corner	5.4.1
`afemStokesMINI_Smooth`	Smooth data	5.4.2
`afemStokesMINI_Smooth2`	Smooth data	5.4.3
`afemStokesMINI_CollidingFlow`	Colliding flow	5.4.4
`afemStokesMINI_BackStep`	Backward facing step	5.4.5
Stokes problems for Crouzeix-Raviart nonconforming finite element method		
`afemStokesCR_Lshape`	Reentrant corner	5.7.1
`afemStokesCR_Smooth`	Smooth, convex domain	5.7.2
`afemStokesCR_Smooth2`	Smooth, convex domain	5.7.3
`afemStokesCR_CollidingFlow`	Smooth, convex domain	5.7.4
`afemStokesCR_BackStep`	Backward facing step	5.7.5
Obstacle problems		
`afemObstacle_Square2`	Square domain	6.4.1
`afemObstacle_Lshape`	Reentrant corner	6.4.2
`afemObstacle_Cusp`	Cusp-shaped obstacle	6.4.3
`afemObstacle_Pyramid`	Pyramid obstacle	6.4.4
`afemObstacle_Square3`	Smooth obstacle	6.4.5

Table A.2: List of benchmark problems of this thesis and the corresponding `MATLAB` files.

```
[x,~,~,~,sigma4e] = solveP1Poisson(f,g,u4Db,alpha,c4n,n4e,n4sDb,n4sNb);
alpha4e = alpha(computeMid4e(c4n,n4e));
[eta4e,eta] = estimateP1EtaB(f,g,sigma4e,c4n,n4e,n4sDb,n4sNb,alpha4e,degree_f,degree_g,0,0);
[~,osc4e] = oscillations(f,c4n,n4e,degree_f);
CP=0.2610;
osc4e = CP^2*osc4e./alpha4e;
eta4e = (sqrt(eta4e) + sqrt(osc4e)).^2;
if ~isempty(n4sNb)
    [~,osc4Nbs,~,e4Nbs] = oscillationsNb(g,c4n,n4e,n4sNb,degree_g);
    CNb=1.1473;
    mid4Nbs = computeMid4s(c4n,n4sNb);
    alpha4Nbs = alpha(mid4Nbs);
    osc4Nbs = CNb^2*osc4Nbs./alpha4Nbs;
    oscNb4e = accumarray(e4Nbs,osc4Nbs,[size(n4e,1) 1]);
    eta4e = (sqrt(eta4e) + sqrt(oscNb4e)).^2;
end
eta = sqrt(sum(eta4e));
[~,etaDb4s,e4p] = DbError(TD2u4Db,c4n,n4e,n4sDb,degree_u4Db);
alpha4p = alpha4e(e4p,:);
CDb = 0.4980;
etaDb4s = CDb^2*alpha4p.*etaDb4s;
etaDb = sqrt(sum(etaDb4s));
etaDb4e = accumarray(e4p,etaDb4s,[size(n4e,1) 1]);
eta4e = eta4e + etaDb4e;
eta = sqrt(eta^2 + etaDb^2);
```

Listing A.2: Example for a call of the Braess error estimator plus overhead terms.

adaptive mesh refinement loop. This might be a nice starter for own experiments.

To compare the error estimator with the exact energy error for problems with known exact solutions, there is the function `exactEnergyError` in the subfolder `integrate`. This function computes $\left\|\mathbb{S}^{-1/2}\nabla_{\mathrm{NC}}(u-u_h)\right\|_{L^2(\Omega)}$. To do so, the function needs the function handle `alpha` and a function handle for ∇u, as well as `sigma4e`, the triangulation data and the degree for the quadrature. This degree was set to 15 for all experiments in this thesis. This means that the energy error for polynomials $u \in \mathcal{P}_{16}(\mathcal{T})$ is integrated exactly.

The next sections give some insight in the implementations of some of the error estimators.

A.3 Implementation of the Braess Equilibration Error Estimator

Listing A.3 displays the code for `estimateP1EtaB` that computes $\left\|\mathbb{S}^{-1/2}(\sigma_h - q_{\mathrm{B}})\right\|_{L^2(\Omega)}$ (the overhead terms like oscillations are computed separately). The tensor σ_h is assumed to be piecewise constant and represented by the $|\mathcal{T}| \times 2$ matrix `sigma4e`. Similarly, `alpha4e` contains the elementwise constant diffusion weight.

The main part of the implementation of the Braess equilibration error estimator is the function `ConstantFluxEquilibrationBraess` in Line 3 of Listing A.3 that solves the local problems in (3.15) and computes the normal fluxes `Fluxes4e` of the broken Raviart-Thomas element $\sigma_h - q_{\mathrm{B}}$. The array `Fluxes4e` has the same size as `s4e` and contains the normal fluxes for $\sigma_h - q_{\mathrm{B}}$ over the three sides for each element, i.e., the coefficients for the local versions of the Raviart-Thomas basis functions. The parameter

Filename	Description	Reference
	Estimators for Poisson and obstacle problems (only $\mathcal{P}_1$ conforming finite element method)	
`estimateP1EtaB`	Braess equilibration estimator η_{B}	Subsection 3.2.3
`estimateP1EtaLS`	Least squares estimator η_{LS}	Subsection 3.2.5
`estimateP1EtaLW`	Luce-Wohlmuth equilibration estimator η_{LW}	Subsection 3.2.2
`estimateP1EtaMFEM`	MFEM equilibration estimator η_{M}	Subsection 3.2.4
`estimateP1EtaR`	Residual-based error estimator η_{R}	Definition 3.4.1
	and refinement indicators (without overhead terms)	Formulas (4.7) and (6.17)
	Estimators for Poisson and Stokes problems (only Crouzeix-Raviart nonconforming finite element method)	
`estimateCREtaA`	Ainsworth interpolation η_{A}	Subsubsection 4.4.2.1
`estimateCREtaAP2`	Ainsworth $\mathcal{P}_2$ Interpolation η_{AP2}	Subsubsection 4.4.2.1
`estimateCREtaARED`	Modified Ainsworth interpolation η_{ARED}	Subsubsection 4.4.2.2
`estimateCREtaMP1`	Optimal $\mathcal{P}_1$ interpolation η_{MP1}	Subsubsection 4.4.2.3
`estimateCREtaMP1RED`	Optimal $\mathcal{P}_1(\mathrm{red}(\mathcal{T}))$ interpolation η_{MP1RED}	Subsubsection 4.4.2.3
`estimateCREtaMP2`	Optimal $\mathcal{P}_2$ interpolation η_{MP2}	Subsubsection 4.4.2.3
`estimateCREtaPMRED`	Patchwise minimal interpolation η_{PMRED}	Subsubsection 4.4.2.2
`estimateCREtaPMRED_Stokes`	As above but modified for Stokes problems	Subsubsection 5.6
`estimateCREtaR`	Explicit Upper bound η_0 for conforming residual	Theorem 4.4.1.(c)
`refinementIndicatorsCR`	refinement indicators (without overhead terms)	Formulas (4.20) and (5.17)
	Estimators for Stokes problems (only mini finite element method)	
`estimateMINIEtaB`	Braess equilibration estimator η_{LW}	Section 5.3
`estimateMINIEtaLW`	Luce-Wohlmuth equilibration estimator η_{LW}	Section 5.3
`estimateMINIEtaMFEM`	MFEM equilibration estimator η_{LW}	Section 5.3
`estimateMINIEtaR`	refinement indicators (without overhead terms)	Formula (5.13)

Table A.3: List of new error estimators in the `AFEM` package and references to detailed descriptions.

```
function [eta4e,eta,Fluxes4e,RTMama] = estimateP1EtaB(f,g,sigma4e,c4n,n4e,n4sDb,n4sNb,alpha4e,...
    degree_f,degree_g,ppiterations,ppredrefinements,meancorrection,f_fine)
[Fluxes4e,RTMama] = ConstantFluxEquilibrationBraess(1,f,g,sigma4e,c4n,n4e,n4sNb,n4sDb,...
    1./alpha4e,degree_f,degree_g);
if ppiterations ~=0
    if strcmp(meancorrection,'true')
        [Fluxes4e,RTMama,parents4e] = PostProcessing(ppiterations,ppredrefinements,...
            c4n,n4e,n4sDb,n4sNb,Fluxes4e,1./alpha4e,[],f_fine,degree_f);
    else
        [Fluxes4e,RTMama,parents4e] = PostProcessing(ppiterations,ppredrefinements,...
            c4n,n4e,n4sDb,n4sNb,Fluxes4e,1./alpha4e);
    end
else
    parents4e = 1:size(n4e,1);
end
eta4e = matMul(matMul(permute(Fluxes4e,[1 3 2]),RTMama),Fluxes4e);
eta4e = accumarray(parents4e(:),eta4e);
eta = sqrt(sum(eta4e));
```

Listing A.3: Listing for `estimateP1etaB.m`

`ppredrefinements` and `ppiterations` indicate the amount of red-refinements and the number of cg iterations for the computation of the Curl v postprocessing (see Section 3.3 for the theoretical details) in Lines 5–12. The output of `Postprocessing` are the fluxes of $\sigma_h - q_{\mathrm{B}} - \operatorname{Curl} v$ and the local Raviart-Thomas mass matrices `RTMama` with respect to a possibly refined triangulation. Therefore, the vector `parents4e` associates the element numbers in the refined triangulation to their parent elements in the original triangulation. Line 16 uses the local Raviart-Thomas mass matrices to compute the norm $\|\sigma_h - q_{\mathrm{B}} - \operatorname{Curl} v\|^2_{L^2(T)}$ for all elements in the refined triangulation. Line 17 accumulates the values of the child elements to their parent elements and the last line computes the total norm.

The presentation of the main subroutine `ConstantFluxEquilibrationBraess` is split into the Listings A.4-A.6. The first part (Listing A.4) prepares all data that is needed for the equilibration process, such as the piecewise integrals of the data f times nodal basis functions in Lines 28–33 and the edgewise integrals of g times nodal basis functions in Lines 35–43. Furthermore, the first part computes the curls of the nodal basis functions and their elementwise normal fluxes in Lines 16–25, as well as the normal fluxes `sigmafluxes4e` (of size $|\mathcal{T}| \times 3$) of the discrete stress σ_h (represented by the array `sigma4e` of size $|\mathcal{T}| \times 2$) in Line 26. Lines 50–51 compute the local Raviart-Thomas mass matrices and multiply them with the (diffusion) weight.

The second part (Listing A.5) calculates the optimal fluxes of the local solution of the local problems (3.15) for boundary nodes $z \in \mathcal{N}(\partial\Omega)$. This is done via a clockwise algorithm in Lines 57–68 that computes one element of $Q(\mathcal{T}(z))$. This involves the sort of the elements of the nodal patch by the subroutine `SortPatch` (see Listing A.6). Then, a constant is added such that either the Neumann boundary conditions are satisfied (then the local solution is unique) or the weighted L^2 norm is minimised. In the latter case any multiplicative of Curl φ_z may be added to the local solution and the subroutine `FluxOptimizer` (see Listing A.6) computes the optimal constant in front of Curl φ_z.

The third part (Listing A.6) solves the local problems (3.15) for the interior nodes in Lines 99–127 in the same manner. Eventually, Lines 128–130 combine all the normal fluxes

```
function [Fluxes4e,RTMama] = ConstantFluxEquilibrationBraess(...
    optimal,f,g,sigma4e,c4n,n4e,n4sNb,n4sDb,weights4e,degree_f,degree_g)
normals4e = computeNormal4e(c4n,n4e);
area4e    = computeArea4e(c4n,n4e);
n4s       = computeN4s(n4e);
length4s  = computeLength4s(c4n,n4s);
e4n       = computeE4n(n4e);
s4e       = computeS4e(n4e);
e4s       = computeE4s(n4e);
s4n       = computeS4n(n4e);
Dbs       = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
nrElems   = size(n4e,1);
nrDbs     = size(Dbs,1);
nrNodes   = size(c4n,1);

curl4e = zeros(size(n4e,1),2,3);
for j = 1 : size(n4e,1)
    nodes = n4e(j,:);
    coords = c4n(nodes,:);
    curl4e(j,:,:) = ([1,1,1;coords']\[0,0;0 -1;1 0])';
end
curlcurl4e = permute(sum(curl4e.*curl4e,2),[1 3 2]).*(area4e.*weights4e*ones(1,3));
curlfluxes4e(1,:,:) = length4s(s4e).*matMul(permute(normals4e,[3 1 2]),curl4e(:,:,1));
curlfluxes4e(2,:,:) = length4s(s4e).*matMul(permute(normals4e,[3 1 2]),curl4e(:,:,2));
curlfluxes4e(3,:,:) = length4s(s4e).*matMul(permute(normals4e,[3 1 2]),curl4e(:,:,3));
sigmafluxes4e = length4s(s4e).*matMul(permute(normals4e,[3 1 2]),sigma4e);
jumps4s = accumarray(s4e(:),sigmafluxes4e(:));
if nargin(f)==3
    integrand=@(n4p,pts,pts_ref)(f(n4p,pts,pts_ref)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)]);
else
    integrand=@(n4p,pts,pts_ref)(f(pts)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)]);
end
fvalues4e = -integrate(c4n,n4e,integrand,degree_f+1);

if ~isempty(n4sNb)
    normals4Nbs = computeNormal4s(c4n,n4sNb);
    integrand=@(n4p,pts,pts_ref)(g(pts,normals4Nbs)*[1-pts_ref pts_ref]);
    gvalues4s = integrate(c4n,n4sNb,integrand,degree_g+1);
    normals4Nbs= computeNormal4s(c4n,n4sNb);
    Nbs = rowaddr(s4n,n4sNb(:,1),n4sNb(:,2));
    elems = nonzeros(diag(e4n(n4sNb(:,1),n4sNb(:,2))))';
    B = sigma4e(elems,:).*normals4Nbs;
    BF4Nbs =(1/2)*length4s(Nbs).*sum(B,2);
else
    Nbs = zeros(0,1);
    gvalues4s = 0;
    BF4Nbs = 0;
end

RTMama = LocalRTMassMatrix(c4n,n4e,area4e);
RTMama = matMul(RTMama,weights4e);
OuterEdges = [n4sDb Dbs; n4sNb Nbs];
OuterEdges(:,4) = e4s(OuterEdges(:,3),1);
EQfluxes = zeros(nrElems,3,3);
sliced_e4n = e4n(:,OuterEdges(:,2));
```

Listing A.4: Listing for Lines 1–55 of `ConstantFluxEquilibrationBraess.m`

```
56 for curEdge = 1:size(OuterEdges,1)
       curNode = OuterEdges(curEdge,2);
58     startelem = OuterEdges(curEdge,4);
       elems = nonzeros(sliced_e4n(:,curEdge));
60     nr = find(elems == startelem);
       elems([1 nr]) = elems([nr 1]);
62     [elems,edges,pos_elems] = SortPatch(elems);
       fluxcount = length(elems)*2;
64     A = eye(fluxcount,fluxcount)-diag(ones(fluxcount-1,1),-1);
       b = zeros(fluxcount,1);
66     b(2:2:fluxcount) = diag(fvalues4e(elems,pos_elems));
       b(3:2:fluxcount-1) = jumps4s(edges(1:end-1))/2;
68     fluxes = A\b;
       rightedge_pos = pos_elems;
70     leftedge_pos = mod(pos_elems+1,3)+1;
       if curEdge > nrDbsges
72         snr = OuterEdges(curEdge,3);
           Nbsnr = find(Nbs == snr);
74         pos = find(n4s(snr,:) == curNode);
           c = BF4Nbs(Nbsnr) - gvalues4s(Nbsnr,pos);
76     elseif ismember(edges(end),Nbs)
           snr = edges(end);
78         Nbsnr = find(Nbs == snr);
           pos = find(n4s(snr,:) == curNode);
80         c = -(gvalues4s(Nbsnr,pos) - BF4Nbs(Nbsnr)) - fluxes(end);
       elseif (optimal == 1)
82         fluxes = fluxes - sum(fluxes)/length(fluxes);
           fluxes4e = zeros(length(elems),3);
84         for cE=1:size(elems,1)
               fluxes4e(cE,rightedge_pos(cE)) = fluxes(cE*2);
86             fluxes4e(cE,leftedge_pos(cE)) = -fluxes(cE*2-1);
           end
88         c = FluxOptimizer(fluxes4e,elems,pos_elems);
       else
90         c = - sum(fluxes)/length(fluxes);

92     end
       for cE=1:length(elems)
94         EQfluxes(elems(cE),rightedge_pos(cE),pos_elems(cE)) = fluxes(cE*2) + c;
           EQfluxes(elems(cE),leftedge_pos(cE),pos_elems(cE)) = -fluxes(cE*2-1) - c;
96     end
   end
```

Listing A.5: Listing for Lines 56–97 of `ConstantFluxEquilibrationBraess.m`

```
98  innernodes = setdiff(1:nrNodes,OuterEdges(:,2))';
    if nnz(innernodes) ~= 0
100     for in = 1:size(innernodes,1)
            curNode = innernodes(in);
102         elems = nonzeros(e4n(:,curNode));
            [elems,edges,pos_elems] = SortPatch(elems);
104         fluxcount = length(elems)*2;
            A = eye(fluxcount,fluxcount)-diag(ones(fluxcount-1,1),-1);
106         b = zeros(fluxcount,1);
            b(2:2:fluxcount) = diag(fvalues4e(elems,pos_elems));
108         b(3:2:fluxcount-1) = jumps4s(edges(1:end-1))/2;
            fluxes = A\b;
110         rightedge_pos = pos_elems;
            leftedge_pos = mod(pos_elems+1,3)+1;
112         if (optimal == 1)
                fluxes4e = zeros(length(elems),3);
114             for cE=1:size(elems,1)
                    fluxes4e(cE,rightedge_pos(cE),1) = fluxes(cE*2);
116                 fluxes4e(cE,leftedge_pos(cE),1) = -fluxes(cE*2-1);
                end
118             c = FluxOptimizer(fluxes4e,elems,pos_elems);
            else
120             c = - sum(fluxes)/length(fluxes);
            end
122         for cE=1:length(elems)
                EQfluxes(elems(cE),rightedge_pos(cE),pos_elems(cE)) = fluxes(cE*2) + c;
124             EQfluxes(elems(cE),leftedge_pos(cE),pos_elems(cE)) = -fluxes(cE*2-1) - c;
            end
126     end
    end
128 Fluxes4e(:,1) = -sum(EQfluxes(:,1,:),3);
    Fluxes4e(:,2) = -sum(EQfluxes(:,2,:),3);
130 Fluxes4e(:,3) = -sum(EQfluxes(:,3,:),3);

132     function [elems,edges,pos_elems] = SortPatch(elems)
            edges = zeros(length(elems),1);
134         pos_elems = zeros(length(elems),1);
            for k=1:length(elems)
136             nodes = n4e(elems(k),:);
                pos_elems(k) = find(nodes == curNode);
138             nextnode = nodes(mod(pos_elems(k),3)+1);
                edges(k) = s4n(curNode,nextnode);
140             if k < length(elems)
                    nextelem = e4n(nextnode,curNode);
142                 nr = find(elems == nextelem);
                    elems([k+1 nr]) = elems([nr k+1]);
144             end
            end
146     end

148     function corrflux = FluxOptimizer(fluxes4e,elems,pos_elems)
            locRTMama = RTMama(elems,:,:);
150         fluxes4curl = zeros(length(elems),3);
            loccurlcurl4e = diag(curlcurl4e(elems,pos_elems));
152         for k=1:length(elems)
                fluxes4curl(k,pos_elems(k)) = curlfluxes4e(pos_elems(k),elems(k),pos_elems(k));
154             fluxes4curl(k,mod(pos_elems(k)+1,3)+1) = ...
                    -curlfluxes4e(mod(pos_elems(k)+1,3)+1,elems(k),mod(pos_elems(k)+1,3)+1);
156         end
            qcurl4e = matMul(matMul(permute(fluxes4e,[1 3 2]),locRTMama),fluxes4curl);
158         qcurl = sum(qcurl4e);
            corrflux = -qcurl*curlfluxes4e(pos_elems(1),elems(1),pos_elems(1))/sum(loccurlcurl4e);
160     end
    end
```

Listing A.6: Listing for Lines 98-end of `ConstantFluxEquilibrationBraess.m`

of the local solutions to the normal fluxes of the (broken) global Raviart-Thomas function $\sigma_h - q_B$.

A.4 Implementation of the Luce-Wohlmuth Equilibration Error Estimator

The Luce-Wohlmuth design from Subsection 3.2.2 is based on the dual triangulation. Recall Listing 3.1 for the computation and enumeration of the triangulation data of the dual refinement with the function `refineDual`.

The presentation of the lengthy `MATLAB` code for the Luce-Wohlmuth error estimator is split into the five Listings A.7-A.11. The first part (Listing A.7) begins with the computation of enumeration arrays, right-hand side integrals in Lines 21–37 (f times nodal basis functions and g times nodal basis functions). Line 38 calls the dual mesh refinement routine and Lines 44–45 compute the normal fluxes `sigmafluxes4e` of the discrete stress σ_h (given through the $|\mathcal{T}| \times 2$ array `sigma4e`) with respect to the dual mesh, i.e., `sigmafluxes4e` is of size $|Tri^\star| \times 3$. Then, the information for the boundary edges are collected in the array `OuterEdges`. The for loop in Line 51-79 computes the normal fluxes of an element from the set $Q(\mathcal{T}^\star(z))$ in the local problems (3.11) for all boundary nodes. This element is unique for boundary patches that include Neumann boundary edges. Before that, the next for loop in Lines 81–109 computes the normal fluxes of an element in the set $Q(\mathcal{T}^\star(z))$ for all interior nodes. The structure of both for loops is similar to the structure of the for loops in the implementation of the Braess equilibration error estimator. The main ingredients are the sort of the patch elements and the solve of a small local linear system of equations by the subroutine `FluxSolver` in Lines 120–157. This solver guarantees that its solution yields a Raviart-Thomas element with correct divergence and Neumann boundary fluxes.

The further patchwise minimisation of the L^2 norm $\left\|\mathbb{S}^{-1/2}(q_{\mathrm{LW}} - \sigma_h)\right\|_{L^2(\omega_z^\star)}$ and the postprocessing is done in the subroutine `LWPostProcessing`. This function computes the normal fluxes of Curl v, for some piecewise linear function $v \in \mathcal{P}_1(\mathcal{T}) \cap C(\Omega)$, which is handled as a broken Raviart-Thomas element. The function v is a linear combination of the nodal basis functions $\varphi_z^\star$ for the nodes of the dual triangulation $z \in \mathcal{N}^\star$. The input parameter `ppiterations` determines how the coefficients of the $\varphi_z^\star$ are computed. The choice `ppiterations = 0` leads to the usual η_{LW}. Here only the coefficients α_z of $\varphi_z^\star$ for the nodes $z \in \mathcal{N}$ of the original mesh are nonzero and chosen such that $\left\|\mathbb{S}^{-1/2}(q_{\mathrm{LW}} - \sigma_h - \alpha_z \operatorname{Curl} \varphi_z^\star)\right\|_{L^2(\omega_z^\star)}$ is minimal. The optimal coefficients are computed in Line 191. For `ppiterations > 0` all basis functions become active and a global linear system of equations is solved approximately with `ppiterations` pcg iterations in Line 195. For `ppiterations = -1` the system is solved exactly in Line 201. The structure of the linear system of equations is explained in Subsection 3.3.2. Note that Neumann boundary nodes have to be excluded in all these modifications, since otherwise q_{LW} might not satisfy the Neumann boundary conditions (the normal fluxes of $\varphi_z^\star$ are nonzero along Γ_N for $z \in \mathcal{N}^\star(\Gamma_N)$). The corresponding node numbers are extracted in Lines 189,194 and 198 for the different cases.

So far, the function computes η_{LW} or the postprocessing $\eta_{\mathrm{LW}(k)}$. For the computation of the mean-corrected version $\eta_{\mathrm{LWm(k)}}$, further modifications are initiated in Lines 111–116.

The subroutines `DivMeanCorrection` in Lines 253–281 (and `NbFluxCorrection` in Lines 212–251) redistribute the divergence (and the normal fluxes) in all child elements (and child edges) of an element $T \in \mathcal{T}$ (and of an Neumann edge $E \in \mathcal{E}(\Gamma_N)$) such that $\operatorname{div} q_{\mathrm{LWm}} = -f_{\mathcal{T}^\star}$ in Ω (and $q_{\mathrm{LWm}} \cdot \nu = g_{\mathcal{E}^\star}$ along Γ_N) as explained in Remark 3.2.5.(c).

Eventually, the array `Fluxes4FineE` contains the normal fluxes of the broken Raviart-Thomas element $q_{\mathrm{LW}} - \sigma_h - \operatorname{Curl} v$ with respect to the dual mesh $\mathcal{T}^\star$ and Line 207 computes its elementwise norm with the help of the weighted Raviart-Thomas mass matrices `RTMama`. See Listing 2.5 and the explanations in Subsubsection 2.3.5.5 for details on the computation of `RTMama`.

A.5 Implementation of the Least-Square Error Estimator

Listing A.12 displays the `MATLAB` code for the realisation of Algorithm 3.1. The input parameter `CF` is the Friedrichs constant $C(\Omega, \Gamma_D)$ for the domain Ω, `beta0` is the initial value for λ in Algorithm 3.1 (default is `beta0=1`), and `LSiterations` is the number of iterations in the for loop of Algorithm 3.1 (default is `LSiterations=3`). Lines 10–15 compute the piecewise integrals `f4e` of f and the normal fluxes of the discrete stress tensor σ_h such that the size of `sigmafluxes4e` matches the size of `s4e`. Lines 17–20 discretise the bilinear forms

$$a(p,q) = \int_\Omega \mathbb{S} p \cdot q \,\mathrm{d}x \quad \text{and} \quad b(p,q) = \int_\Omega \operatorname{div} p \operatorname{div} q \,\mathrm{d}x$$

by computation of the elementwise matrices

$$\texttt{BT(m,j,k)} = \int_{T_m} \mathbb{S}\vartheta_{E_j} \cdot \vartheta_{E_k} \,\mathrm{d}x \quad \text{and} \quad \texttt{CT(m,j,k)} = \int_{T_m} \operatorname{div}(\vartheta_{E_j}) \operatorname{div}(\vartheta_{E_k}) \,\mathrm{d}x$$

for the three edges $E_1, \ldots, E_3 = \mathcal{E}(T_m)$ of the m-th triangle $T_m \in \mathcal{T}$.

The first matrix `BT` contains the local weighted Raviart-Thomas mass matrices `RTMama` from Line 17 up to the correct signs of the basis functions on each triangle. The signs are fixed in Lines 18–19, confer to Subsubsection 2.3.5.5 for more details on the choice of the orientation and the computation of the signs by `computeSig4e`. The second matrix `CT` is very easy to calculate, because the divergence of the all Raviart-Thomas basis functions equals $\operatorname{div} \vartheta_{E_j} = \nu_{T_m} \nu_{E_j} 1/|T_m|$ for $j = 1, \ldots, 3$. Hence,

$$\texttt{CT(m,j,k)} = \nu_{E_j} \cdot \nu_{E_k} / |T_m| = \texttt{sig4e(m,j)*sig4e(m,k)/area4e(m).}$$

This is calculated in Line 20. Next, the components of the right-hand side vector are computed. The vector `bP1` contains the integrals of σ_h times Raviart-Thomas basis functions. These are computed with the help of the Raviart-Thomas mass matrices and the fluxes of σ_h that are computed in Line 15. The entry `bf(m,j)` of the vector in Line 23 contains the integrals of f times $\operatorname{div} \vartheta_{E_j}$ over T_m, i.e.,

$$\texttt{bf(m,j)} = \nu_{E_j} \cdot \nu_{T_m} f_{T_m} = \texttt{sig4e(m,j)*f4e(m)./area4e(m).}$$

```
function [eta4e,eta] = estimateP1EtaLW(f,g,sigma4e,c4n,n4e,n4sDb,n4sNb,alpha4e,degree_f,degree_g,...
                                       ppiterations,meancorrection,f_fine,res4n)
if nargin < 8, alpha4e = ones(size(n4e,1),1); end
if nargin < 9, degree_f = 1; end
if nargin < 10, degree_g = 1; end
if nargin < 11, ppiterations = 1; end
if nargin < 12, meancorrection = 'false'; end
if nargin < 13, f_fine = f; end
if nargin < 14, res4n = zeros(size(c4n,1),1); end
e4n      = computeE4n(n4e);
e4s      = computeE4s(n4e);
s4n      = computeS4n(n4e);
n4s      = computeN4s(n4e);
area4e   = computeArea4e(c4n,n4e);
Dbs      = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
Nbs      = rowaddr(s4n,n4sNb(:,1),n4sNb(:,2));
nrElems  = size(n4e,1);
nrDbsges = size(Dbs,1);
nrNodes  = size(c4n,1);

if isa(f,'function_handle')
    if nargin(f)==3
        integrand=@(n4p,pts,pts_ref)(f(n4p,pts,pts_ref)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)]);
    else
        integrand=@(n4p,pts,pts_ref)(f(pts)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)]);
    end
    fvalues4e = integrate(c4n,n4e,integrand,degree_f+1);
else
    fvalues4e = f;
end
if ~isempty(n4sNb)
    normals4Nbs = computeNormal4s(c4n,n4sNb);
    integrand=@(n4p,pts,pts_ref)(g(pts,normals4Nbs)*[1-pts_ref pts_ref]);
    gvalues4s = integrate(c4n,n4sNb,integrand,degree_g+1);
else
    gvalues4s = 0;
end
[c4n_fine,n4e_fine,n4sDb_fine,n4sNb_fine,parents4e,pos_pelems] = refineDual(c4n,n4e,n4sDb,n4sNb);
s4e_fine = computeS4e(n4e_fine);
n4s_fine = computeN4s(n4e_fine);
normals4e_fine = computeNormal4e(c4n_fine,n4e_fine);
length4s_fine = computeLength4s(c4n_fine,n4s_fine);
area4e_fine = area4e(parents4e)/6;
prefluxes = matMul(permute(normals4e_fine,[3 1 2]),sigma4e(parents4e,:));
sigmafluxes4e = length4s_fine(s4e_fine).*prefluxes;
OuterEdges = [n4sDb Dbs; n4sNb Nbs];
OuterEdges(:,4) = e4s(OuterEdges(:,3),1);
nrOutEdges = size(OuterEdges,1);
Fluxes4FineE = zeros(nrElems*6,3);

for curEdge=1:nrOutEdges
    curNode = OuterEdges(curEdge,2);
    startelem = OuterEdges(curEdge,4);
    startedge = OuterEdges(curEdge,3);
    elems = nonzeros(e4n(:,curNode));
    edges = zeros(length(elems)+1,1);
    edges(1) = startedge;
    pos_elems = zeros(length(elems),1);
    nr = find(elems == startelem);
    elems([1 nr]) = elems([nr 1]);
    for j=1:length(elems)
        nodes = n4e(elems(j),:);
        pos_elems(j) = find(nodes == curNode);
        nextnode = nodes(mod(pos_elems(j),3)+1);
        edges(j+1) = s4n(curNode,nextnode);
```

Listing A.7: Listing for Lines 1–65 of `estimateP1EtaLW.m`

```
66          if j < length(elems)
                nextelem = e4n(nextnode,curNode);
68              nr = find(elems == nextelem);
                elems([j+1 nr]) = elems([nr j+1]);
70          end
        end
72      global_fine_elems = zeros(length(elems)*2,1);
        global_fine_elems(1:2:end-1) = (elems-1)*6+(pos_elems-1)*2+1;
74      global_fine_elems(2:2:end) = (elems-1)*6+(pos_elems-1)*2+2;
        [top_fluxes,inner_fluxes]= FluxSolver(curNode,global_fine_elems,edges,true);
76      fluxes4e = [inner_fluxes(2:fluxcount) top_fluxes(1:end) -inner_fluxes(1:fluxcount-1)];
        fluxes4e = fluxes4e - sigmafluxes4e(global_fine_elems,:);
78      Fluxes4FineE(global_fine_elems,:) = fluxes4e;
    end
80  innernodes = setdiff(1:nrNodes,OuterEdges(:,2))';
    if nnz(innernodes) ~= 0
82      for in=1:size(innernodes,1)
            curNode = innernodes(in);
84          elems = nonzeros(e4n(:,curNode));
            edges = zeros(length(elems)+1,1);
86          pos_elems = zeros(length(elems),1);
            for j=1:length(elems)
88              nodes = n4e(elems(j),:);
                pos_elems(j) = find(nodes == curNode);
90              nextnode = nodes(mod(pos_elems(j),3)+1);
                edges(j+1) = s4n(curNode,nextnode);
92              if j < length(elems)
                    nextelem = e4n(nextnode,curNode);
94                  nr = find(elems == nextelem);
                    elems([j+1 nr]) = elems([nr j+1]);
96              end
            end
98          edges(1) = edges(end);
            global_fine_elems = zeros(length(elems)*2,1);
100         global_fine_elems(1:2:end-1) = (elems-1)*6+(pos_elems-1)*2+1;
            global_fine_elems(2:2:end) = (elems-1)*6+(pos_elems-1)*2+2;
102         [top_fluxes,inner_fluxes]=FluxSolver(curNode,global_fine_elems,edges,false);
            fluxcount = length(top_fluxes);
104         fluxes4e = [[inner_fluxes(2:fluxcount); inner_fluxes(1)] top_fluxes(1:end)...
                -inner_fluxes(1:fluxcount)];
106         fluxes4e = fluxes4e - sigmafluxes4e(global_fine_elems,:);
            Fluxes4FineE(global_fine_elems,:) = fluxes4e;
108     end
    end
110
    if strcmp(meancorrection,'true')
112    Fluxes4FineE = DivMeanCorrection(Fluxes4FineE,c4n_fine,n4e_fine,f_fine,degree_f);
       if ~isempty(n4sNb)
114       Fluxes4FineE = NbFluxCorrection(Fluxes4FineE,c4n_fine,n4e_fine,n4sNb,n4sNb_fine,g,degree_g);
       end
116 end
    [eta4e,eta,Fluxes4FineE,RTMama]=LWPostProcessing(ppiterations,c4n,n4e_fine,c4n_fine,n4sNb_fine,...
118     Fluxes4FineE,1./alpha4e(parents4e));

120     function [top_fluxes,inner_fluxes]=FluxSolver(curNode,fine_elems,edges,boundary)
            loc_pelems = parents4e(fine_elems);
122         loc_pos_pelems = pos_pelems(fine_elems);
            top_fluxes = sigmafluxes4e(fine_elems,2);
124         PQf4e = diag(fvalues4e(loc_pelems(1:end),loc_pos_pelems(1:end)))/2...
                - (res4n(curNode)/sum(area4e_fine(fine_elems)))*area4e_fine(fine_elems);
126         if (boundary == true)
                fluxcount = length(fine_elems)+1;
128             first_flux = -sigmafluxes4e(fine_elems(1),3);
                last_flux = sigmafluxes4e(fine_elems(end),1);
130             outfluxsum = sum(top_fluxes) - first_flux + last_flux;
```

Listing A.8: Listing for Lines 66–130 of `estimateP1EtaLW.m`

```
            PQf_raw = sum(diag(fvalues4e(loc_pelems(1:2:end),loc_pos_pelems(1:2:end))));
132         alphaK = (-PQf_raw-outfluxsum);
            if curEdge > nrDbsges
134             snr = OuterEdges(curEdge,3);
                Nbsnr = find(Nbs == snr);
136             pos = find(n4s(snr,:) == curNode);
                startvalue = -gvalues4s(Nbsnr,pos);
138         else
                if ismember(edges(end),Nbs)
140                 snr = edges(end);
                    Nbsnr = find(Nbs == snr);
142                 pos = find(n4s(snr,:) == curNode);
                    startvalue = first_flux-last_flux-gvalues4s(Nbsnr,pos)-alphaK;
144             else
                    startvalue = first_flux - 1/2*alphaK;
146             end
            end
148     else
            fluxcount = length(fine_elems);
150         startvalue = 0;
        end
152     A = eye(fluxcount,fluxcount)-diag(ones(fluxcount-1,1),-1);
        b = zeros(fluxcount,1);
154     b(1) = startvalue;
        b(2:fluxcount) = -top_fluxes(1:fluxcount-1) - PQf4e(1:fluxcount-1);
156     inner_fluxes = A\b;
    end
158
    function [eta4e,eta,quh,RTMama]=LWPostProcessing(ppiterations,c4n,n4e_fine,c4n_fine,...
160                                                  n4sNb_fine,quh,alpha4e_fine)
        A_loc = zeros(3,3,size(n4e_fine,1));
162     curls4e = zeros(size(n4e_fine,1),2,3);
        for j=1:size(n4e_fine)
164         nodes = n4e_fine(j,:);
            coords = c4n_fine(nodes,:);
166         grads = [1,1,1;coords'] \ [0,0;eye(2)];
            curls = [-grads(:,2) grads(:,1)];
168         curls4e(j,:,:) = curls';
            A_loc(:,:,j) = area4e_fine(j)*alpha4e_fine(j)*(curls*curls');
170     end
        n4eT = n4e_fine';
172     I = [n4eT;n4eT;n4eT];
        J = [n4eT(:),n4eT(:),n4eT(:)]';
174     A = sparse(I(:),J(:),A_loc(:));
        RTMama = LocalRTMassMatrix(c4n_fine,n4e_fine,area4e_fine);
176     RTMama = matMul(RTMama,alpha4e_fine);
        fluxes4curl(:,:,1) = length4s_fine(s4e_fine).*matMul(permute(normals4e_fine,[3 1 2]),...
178         curls4e(:,:,1));
        fluxes4curl(:,:,2) = length4s_fine(s4e_fine).*matMul(permute(normals4e_fine,[3 1 2]),...
180         curls4e(:,:,2));
        fluxes4curl(:,:,3) = length4s_fine(s4e_fine).*matMul(permute(normals4e_fine,[3 1 2]),...
182         curls4e(:,:,3));
        b_loc(:,1) = matMul(matMul(permute(quh,[1 3 2]),RTMama),fluxes4curl(:,:,1));
184     b_loc(:,2) = matMul(matMul(permute(quh,[1 3 2]),RTMama),fluxes4curl(:,:,2));
        b_loc(:,3) = matMul(matMul(permute(quh,[1 3 2]),RTMama),fluxes4curl(:,:,3));
186     b = accumarray(n4e_fine(:),-b_loc(:));
        x = zeros(size(c4n_fine,1),1);
188     if ppiterations >= 0
            freeNodes = setdiff(1:size(c4n,1),unique(n4sNb_fine));
190         if ~isempty(freeNodes)
                x(freeNodes) = b(freeNodes)./diag(A(freeNodes,freeNodes));
192         end
            if ppiterations > 0
194             freeNodes = setdiff(1:size(c4n_fine,1),unique(n4sNb_fine));
                [x] = global_pcg(A,b,freeNodes,x,ppiterations);
```

Listing A.9: Listing for Lines 131–195 of `estimateP1EtaLW.m`

```
196             end
        elseif ppiterations == -1
198             freeNodes = setdiff(1:size(c4n_fine,1),unique(n4sNb_fine));
                if ~isempty(freeNodes)
200                 warning off;
                    x(freeNodes) = A(freeNodes,freeNodes)\b(freeNodes);
202             end
        else error('postprocessing: ppiterations must be be greater or equal than -1');
204     end
        Curlx_fluxes4e = matMul(fluxes4curl,x(n4e_fine));
206     quh = quh + Curlx_fluxes4e;
        eta4e_fine = matMul(matMul(permute(quh,[1 3 2]),RTMama),quh);
208     eta4e = accumarray(parents4e(:),eta4e_fine(:));
        eta = sqrt(sum(eta4e_fine));
210 end

212 function quh = NbFluxCorrection(quh,c4n_fine,n4e_fine,n4sNb,n4sNb_fine,g_fine,degree_g)
        e4Nbs = e4s(Nbs,1);
214     s4e_fine = computeS4e(n4e_fine);
        n4s_fine = computeN4s(n4e_fine);
216     s4n_fine = computeS4n(n4e_fine,n4s_fine);
        if nargin(g_fine)==2
218         normals4Nbs_fine = computeNormal4s(c4n_fine,n4sNb_fine);
            integrand=@(n4p,pts,pts_ref)g_fine(pts,normals4Nbs_fine);
220     else
            integrand=@(n4p,pts,pts_ref)g_fine(pts);
222     end
        Nbs_fine = rowaddr(s4n_fine,n4sNb_fine(:,1),n4sNb_fine(:,2));
224     mean4Nbs_fine(Nbs_fine) = integrate(c4n_fine,n4sNb_fine,integrand,degree_g);
        if nargin(g) == 2
226         normals4Nbs = computeNormal4s(c4n,n4sNb);
            integrand=@(n4p,pts,pts_ref)(g(pts,normals4Nbs)*[1-pts_ref pts_ref]);
228     else
            integrand=@(n4p,pts,pts_ref)(g(pts)*[1-pts_ref pts_ref]);
230     end
        g4Nbs = integrate(c4n,n4sNb,integrand,degree_g+1);
232     cnr4Nbs = size(c4n,1) + size(n4e,1) + Nbs;
        A1 = full(s4n_fine(n4sNb(:,1),cnr4Nbs));
234     A2 = full(s4n_fine(cnr4Nbs,n4sNb(:,2)));
        childedges(:,1) = diag(A1);
236     childedges(:,2) = diag(A2);
        for j=1:size(n4sNb,1)
238        nodes = n4e(e4Nbs(j),:);
           k = setdiff(nodes,n4sNb(j,:));
240        switch (find(nodes == k))
               case 3; subelems = [2 3];
242            case 1; subelems = [4 5];
               case 2; subelems = [6 1];
244        end
           subelems = (e4Nbs(j)-1)*6+subelems;
246        quh(subelems(1),1) = quh(subelems(1),1) - g4Nbs(j,1) + mean4Nbs_fine(childedges(j,1));
           quh(subelems(1),2) = quh(subelems(1),2) + g4Nbs(j,1) - mean4Nbs_fine(childedges(j,1));
248        quh(subelems(2),3) = quh(subelems(2),3) - g4Nbs(j,2) + mean4Nbs_fine(childedges(j,2));
           quh(subelems(2),2) = quh(subelems(2),2) + g4Nbs(j,2) - mean4Nbs_fine(childedges(j,2));
250     end
    end
252

    function quh = DivMeanCorrection(quh,c4n,n4e,f_fine,degree_f)
254     if nargin(f_fine)==1
            integrand=@(n4p,pts,pts_ref)f_fine(pts);
256     else
            integrand=f_fine;
258     end
        mean4e = integrate(c4n,n4e,integrand,degree_f);
260     div4e = sum(quh,2);
```

Listing A.10: Listing for Lines 196–260 of `estimateP1EtaLW.m`

```
        diff4e = -(mean4e + div4e);
        corrflux(:,1) = diff4e(1:6:end);
        corrflux(:,2) = diff4e(2:6:end);
        corrflux(:,3) = diff4e(3:6:end);
        corrflux(:,4) = diff4e(4:6:end);
        corrflux(:,5) = diff4e(5:6:end);
        corrflux(:,6) = diff4e(6:6:end);
        quh(6:6:end,3,1) = quh(6:6:end,3,1) + corrflux(:,6,:);
        quh(5:6:end,1,1) = quh(5:6:end,1,1) - corrflux(:,6,:);
        quh(5:6:end,2,1) = quh(5:6:end,2,1) + corrflux(:,6,:) + corrflux(:,5,:);
        quh(4:6:end,2,1) = quh(4:6:end,2,1) - corrflux(:,6,:) - corrflux(:,5,:);
        quh(4:6:end,3,1) = quh(4:6:end,3,1) + corrflux(:,6,:) + corrflux(:,5,:) + corrflux(:,4,:);
        quh(3:6:end,1,1) = quh(3:6:end,1,1) - corrflux(:,6,:) - corrflux(:,5,:) - corrflux(:,4,:);
        quh(3:6:end,2,1) = quh(3:6:end,2,1) + corrflux(:,6,:) + corrflux(:,5,:) + corrflux(:,4,:)...
            + corrflux(:,3,:);
        quh(2:6:end,2,1) = quh(2:6:end,2,1) - corrflux(:,6,:) - corrflux(:,5,:) - corrflux(:,4,:)...
            - corrflux(:,3,:);
        quh(2:6:end,3,1) = quh(2:6:end,3,1) + corrflux(:,6,:) + corrflux(:,5,:) + corrflux(:,4,:)...
            + corrflux(:,3,:) + corrflux(:,2,:);
        quh(1:6:end,1,1) = quh(1:6:end,1,1) - corrflux(:,6,:) - corrflux(:,5,:) - corrflux(:,4,:)...
            - corrflux(:,3,:) - corrflux(:,2,:);
    end
end
```

Listing A.11: Listing for Lines 261–325 of `estimateP1EtaLW.m`

The for loop for the least-square minimisation begins in Line 41. In Line 42, the global system matrix `A` for the least-square problem is accumulated from the local matrices `BT` and `CT` and weighted with `beta(end)` (the current value for λ from Algorithm 3.1). The same is done for the right-hand side vector `b` in Line 43 and the local structures `bf` and `bP1`. The contributions of `bf` get the additional weight `CF` that contains the Friedrichs constant and the correct diffusion weight. As explained in Section 3.2.5 the Neumann fluxes are fixed by $q_{\mathrm{LS}} \cdot \nu = g_{\mathcal{E}}$ along Γ_N and this happens in Lines 45. Finally, the linear system of equations is solved and `x` contains the coefficients for the Raviart-Thomas basis functions. Line 50 computes the elementwise normal fluxes of the difference to σ_h (which is a broken Raviart-Thomas element). Then, the two norms in the majorant $\widehat{M}$ from Algorithm 3.1 are computed (without the oscillations of f and g, which can be computed separately) and λ is updated in Line 55. If the linear system is ill-conditioned or in case there was a division by zero, the loop stops and continues with the previous value for λ. The condition `condest(A(freeSides,freeSides))*eps^0.5>1` was found through several undisplayed experiments. For larger exponents of `eps` the efficiency indices got worse on very fine meshes. Hence, this parameter needs careful consideration. The remaining lines compute the postprocessing of the computed fluxes and the total error `eta`. The output parameter `eqrror` contains the quantity $\|f + \operatorname{div} q_{\mathrm{LS}}\|_{L^2(\Omega)}$.

A.6 Implementation of the AP2 Design

The function `estimateCREtaAP2` of Listing A.14 computes the error estimator $\mu_{\mathrm{AP2}} = \|\!|\!| u_{\mathrm{CR}} - v_{\mathrm{AP2}} |\!|\!|_{\mathrm{NC}}$ of Subsubsection 4.4.2.1 for the Crouzeix-Raviart finite element method for the Poisson problem.

```
function [eta4e,eta,Fluxes4e,RTMama,beta,eqerror] = estimateP1EtaLS(f,g,sigma4e,c4n,n4e,...
    n4sDb,n4sNb,alpha4e,degree_f,degree_g,CF,beta0,LSiterations,ppiterations,ppredrefinements)
normals4e = computeNormal4e(c4n,n4e);
n4s       = computeN4s(n4e);
length4s  = computeLength4s(c4n,n4s);
s4e       = computeS4e(n4e);
area4e    = computeArea4e(c4n,n4e);
sig4e     = computeSig4e(n4e);
nrSides   = size(n4s,1);
if isa(f,'function_handle') && nargin(f) == 3
    f4e = integrate(c4n,n4e,f,degree_f);
elseif isa(f,'function_handle')
    f4e = integrate(c4n,n4e,@(n4p,pts,pts_ref)f(pts),degree_f);
else, f4e = f; end
sigmafluxes4e = length4s(s4e(:,:)).*matMul(permute(normals4e,[3 1 2]),sigma4e);
dofSigma4e = s4e;
RTMama = matMul(LocalRTMassMatrix(c4n,n4e),1./alpha4e);
LT(:,1,1) = sig4e(:,1); LT(:,2,2) = sig4e(:,2); LT(:,3,3) = sig4e(:,3);
BT = matMul(matMul(LT,RTMama),LT);
CT = matMul(matMul(sig4e,permute(sig4e,[1 3 2])),1./area4e);
tmpfluxes4e = sig4e.*sigmafluxes4e;
bP1 = matMul(BT,tmpfluxes4e);
bf = sig4e.*((f4e./area4e)*ones(1,3));
dofSigma4e2 = dofSigma4e';
I = repmat(dofSigma4e2(:),1,size(dofSigma4e,2))';
J = repmat(dofSigma4e,1,size(dofSigma4e,2))';
if ~isempty(n4sNb)
    s4n = computeS4n(n4e);
    Nbs = rowaddr(s4n,n4sNb(:,1),n4sNb(:,2));
    normals = computeNormal4s(c4n,n4sNb);
    if nargin(g) == 1
      g4Nbs = integrate(c4n,n4sNb,@(n4p,pts,pts_ref)g(pts),degree_g);
    else
      g4Nbs = integrate(c4n,n4sNb,@(n4p,pts,pts_ref)g(pts,normals),degree_g);
    end
    freeSides = setdiff(1:nrSides,Nbs);
else, freeSides = 1:nrSides; end
BT = permute(BT,[2 3 1]); CT = permute(CT,[2 3 1]);
CF = CF/sqrt(min(alpha4e));
beta = [beta0];
for j=1:LSiterations
    A = sparse(I(:),J(:),(1+beta(end))*BT(:) + (1+1/beta(end))*CF^2*CT(:));
    b = accumarray(dofSigma4e(:),-(1+1/beta(end))*CF^2*bf(:) + (1+beta(end))*bP1(:));
    x = zeros(nrSides,1);
    if ~isempty(n4sNb), x(Nbs) = g4Nbs; b = b - A*x; end
    warn_state_1 = warning('off','MATLAB:nearlySingularMatrix');
    warn_state_2 = warning('off','MATLAB:singularMatrix');
    x(freeSides) = A(freeSides,freeSides)\b(freeSides);
    warning(warn_state_2); warning(warn_state_1);
    Fluxes4e = sigmafluxes4e - sig4e.*x(s4e);
    etaQ = matMul(matMul(permute(Fluxes4e,[1 3 2]),RTMama),Fluxes4e);
    divergence4e = sum(sig4e.*x(s4e),2);
    eta_fterm = (f4e + divergence4e).^2;
    eqerror = sqrt(CF^2*sum(eta_fterm));
    beta = [beta; eqerror/sqrt(sum(etaQ))];
    if (isnan(beta(end)) || condest(A(freeSides,freeSides))*eps^0.5>1)
        if isnan(beta(end)), beta = beta(1:end-1); end;
        break;
    end
end
if ppiterations ~=0
    [Fluxes4e,RTMama,parents4e]=PostProcessing(ppiterations,ppredrefinements,c4n,n4e,n4sDb,n4sNb,...
                                               Fluxes4e,1./alpha4e);
else, parents4e=1:size(n4e,1); end
eta4e = matMul(matMul(permute(Fluxes4e,[1 3 2]),RTMama),Fluxes4e);
eta4e = accumarray(parents4e(:),eta4e); beta = beta(end);
eta = sqrt((1+beta)*sum(eta4e) + (1+1/beta)*eqerror^2);
```

Listing A.12: Listing of `estimateP1EtaLS.m`

```
function [u_M4e,u_P2] = AinsworthPostProcessing(c4n,n4e,u_CRP24e,mean4e,alpha4e)
nrNodes = size(c4n,1);
s4e = computeS4e(n4e);
mid4e = computeMid4e(c4n,n4e);
n4s = computeN4s(n4e);
nrSides = size(n4s,1);
mid4s = computeMid4s(c4n,n4s);
length4s = computeLength4s(c4n,n4s);
u_M4e(:,1) = u_CRP24e(:,1) - mean4e.*sum((c4n(n4e(:,1),:)-mid4e).^2,2)./alpha4e/4;
u_M4e(:,2) = u_CRP24e(:,2) - mean4e.*sum((c4n(n4e(:,2),:)-mid4e).^2,2)./alpha4e/4;
u_M4e(:,3) = u_CRP24e(:,3) - mean4e.*sum((c4n(n4e(:,3),:)-mid4e).^2,2)./alpha4e/4;
u_M4e(:,4) = u_CRP24e(:,4) - mean4e.*sum((mid4s(s4e(:,1),:)-mid4e).^2,2)./alpha4e/4;
u_M4e(:,5) = u_CRP24e(:,5) - mean4e.*sum((mid4s(s4e(:,2),:)-mid4e).^2,2)./alpha4e/4;
u_M4e(:,6) = u_CRP24e(:,6) - mean4e.*sum((mid4s(s4e(:,3),:)-mid4e).^2,2)./alpha4e/4;
c4e = sum(length4s(s4e).^2,2)./alpha4e/36;
u_M4e = u_M4e + (mean4e.*c4e)*ones(1,6);
if nargout > 1
    u_M4e_nodes = u_M4e(:,1:3);
    u_M4e_sides = u_M4e(:,4:6);
    weight4e = repmat(sqrt(alpha4e),[1 3]);
    u_P2(nrNodes+1:nrNodes+nrSides) = accumarray(s4e(:),u_M4e_sides(:).*weight4e(:))...
        ./accumarray(s4e(:),weight4e(:));
    u_P2(1:nrNodes) = accumarray(n4e(:),u_M4e_nodes(:).*weight4e(:))...
        ./accumarray(n4e(:),weight4e(:));
end
end
```

Listing A.13: Listing of `AinsworthPostProcessing` (subfunction of `estimateCREtaAP2.m`)

The most important part is the subroutine `AinsworthPostProcessing` from Listing A.13. This function computes an array of coefficients `u_M4e` of size $|\mathcal{T}| \times 6$ for the piecewise quadratic function $v^0 \in \mathcal{P}_2(\mathcal{T})$ with respect to the $\mathcal{P}_2$ basis functions. The first three columns `uM4e(:,1:3)` contain the coefficients for the basis function with respect to the nodes `n4e` and the last three columns `uM4e(:,4:6)` contain the coefficients for the basis function with respect to the sides `s4e`. The vector `u_P2` of size $(|\mathcal{N}| + |\mathcal{E}|) \times 1$ contains the basis coefficients of the conforming quadratic interpolation $v_{\mathrm{AP2}} \in \mathcal{P}_2(\mathcal{T}) \cap C(\Omega)$. The input data consists of the triangulation data, the discrete Crouzeix-Raviart solution `u_CRP24e` (as an $3 \times |\mathcal{T}|$ array with the nodal values of u_{CR} on each triangle). Furthermore `mean4e` contains the integral means of the right-hand side data f and `alpha4e` contains the elementwise constant scalar diffusion weights.

Lines 9–16 of Listing A.13 compute `u_M4e` by evaluation of v_0 in the vertices and edge midpoints of every triangle. The constant $f_T(y - \mathrm{mid}(T))^T \mathrm{S}^{-1}(y - \mathrm{mid}(T))\,\mathrm{d}y$ is computed separately in Line 15 with the help of Lemma 2.2.19. The remaining Lines 18–24 compute the averaged values `u_P2` of v_{AP2}. Note that, by identity (4.22) of Section 4.7, `u_M4e` might also be used to evaluate the Raviart-Thomas finite element solution q_{RT} for the right-hand side f and the nodal values `u_CRP24e` of the Crouzeix-Raviart function for the right hand side $f_{\mathcal{T}}$.

The function `estimateCERtaAP2` calls `AinsworthPostProcessing` in Line 29 of Listing A.14. The lines prior to that prepare the input data. Lines 22–28 compute the elementwise nodal values of $u_{\mathrm{CR}} \in \mathcal{P}_1(\mathcal{T})$. The loop in Lines 37–58 computes the energy error of $u_{\mathrm{CR}} - v_{\mathrm{AP2}}$ elementwise with the help of the local $\mathcal{P}_2$ stiffness matrices that are discussed in Subsection 2.3.5.4 and Listing 2.3.

```
function [eta4e,eta] = estimateCREtaAP2(f,u4Db,alpha4e,x,c4n,n4e,n4sDb,n4sNb,degree_f)
area4e = computeArea4e(c4n,n4e);
n4s = computeN4s(n4e);
mid4s = computeMid4s(c4n,n4s);
s4n = computeS4n(n4e);
Dbs = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
s4e = computeS4e(n4e);
nrElems = size(n4e,1);
nrNodes = size(c4n,1);
nrComp  = size(x,2);
if nargin(f) == 1
    f = @(n4p,Gpts4p,Gpts4ref)(f(Gpts4p));
end
f4e = integrate(c4n, n4e, f,degree_f);
c4n_P2 = [c4n;mid4s];
dof4e_P2 = [n4e nrNodes+s4e];
fixedNodes = [unique(n4sDb(:)); nrNodes+Dbs];
val4fixedNodes = u4Db(c4n_P2(fixedNodes,:));
uCR_P24e = zeros(nrElems,6,nrComp);
x_P24e = zeros(nrElems,6,nrComp);
for k=1:nrComp
    xk = x(:,k);
    W = xk(s4e)';
    Z = (ones(3)-2*eye(3))*W;
    uCR_P24e(:,1:3,k) = Z([2 3 1],:)';
    uCR_P24e(:,4,k) = sum(uCR_P24e(:,[1 2]),2)/2;
    uCR_P24e(:,5,k) = sum(uCR_P24e(:,[2 3]),2)/2;
    uCR_P24e(:,6,k) = sum(uCR_P24e(:,[3 1]),2)/2;
    [~,Iux4nk] = AinsworthPostProcessing(c4n,n4e,uCR_P24e(:,:,k),f4e(:,k)./area4e,alpha4e);
    Iux4nk(fixedNodes) = val4fixedNodes(:,k);
    x_P24e(:,:,k) = Iux4nk(dof4e_P2);
end
xdiff4e = uCR_P24e - x_P24e;

eta4e = zeros(nrElems,1);
A_loc = zeros(6,6,nrElems);
for elem = 1 : nrElems
    nodes = n4e(elem,:);
    coords = c4n(nodes,:);
    area = area4e(elem);
    grads = [1 1 1;coords']\[0 0;eye(2)];
    AlocalP1 = alpha4e(elem) * area * grads * grads';
    AlocalP2(1:3,1:3) = AlocalP1.*(4*eye(3)-1);
    Atmp = 4*ones(3,1)*AlocalP1([6 3 2]);
    Atmp = Atmp-diag(diag(Atmp));
    AlocalP2(1:3,[5 6 4]) = Atmp;
    AlocalP2([5 6 4],1:3) = Atmp';
    d = diag(AlocalP1);
    d = d+d([2 3 1])+AlocalP1([2 6 3])';
    Atmp2 = diag(d);
    Atmp2([2 3 6]) = AlocalP1([3 6 2]);
    Atmp2 = Atmp2 + Atmp2' - diag(diag(Atmp2));
    AlocalP2(4:6,4:6) = 8*Atmp2;
    AlocalP2 = AlocalP2/3;
    A_loc(:,:,elem) = AlocalP2;
    xdiff = permute(xdiff4e(elem,:,:),[3 2 1]);
    eta4e(elem) = sum(diag(xdiff*A_loc(:,:,elem)*xdiff'));
end
eta = sqrt(sum(eta4e));
end
```

Listing A.14: Listing of `estimateCREtaAP2.m`

```
label=lst:estimateCREtaPMRED_Stokes,
caption=Listing of \lstinline|estimateCREtaPMRED_Stokes.m|]
function [eta4e,eta,div4e,div]=estimateCREtaPMRED_Stokes(u4Db,alpha4e,x,sigma4e,...
                                                         c4n,n4e,n4sDb,n4sNb,c0,nrSwings)
n4s = computeN4s(n4e); s4e = computeS4e(n4e);
nrNodes = size(c4n,1); nrElems = size(n4e,1); nrSides = size(n4s,1);
[c4n_fine,n4e_fine,n4sDb_fine,~,parents4e] = refineUniformRed(c4n,n4e,n4sDb,n4sNb);
nrFineNodes = size(c4n_fine,1);
area4e_fine = computeArea4e(c4n_fine,n4e_fine);
x1 = x(:,1); x2 = x(:,2);
W = [x1(s4e)' x2(s4e)'];
Z = (ones(3)-2*eye(3))*W;
Iux4n(nrNodes+1:nrNodes+nrSides,:) = [x1 x2];
edgepermute = [2 3 1];
x4e_fine1 = zeros(size(n4e_fine,1),3);
x4e_fine2 = zeros(size(n4e_fine,1),3);
A_loc = zeros(6,6,size(n4e_fine,1));
B_loc = zeros(6,6,size(n4e_fine,1));
b = zeros(2*nrFineNodes,1);
for j=1:size(n4e_fine,1)
    nodes = n4e_fine(j,:);
    [~,J] = find(nodes > nrNodes);
    x4e_fine1(j,J) = x1(nodes(J)-nrNodes);
    x4e_fine2(j,J) = x2(nodes(J)-nrNodes);
    [~,J] = find(nodes <= nrNodes);
    x4e_fine1(j,J) = Z(edgepermute(J),parents4e(j));
    x4e_fine2(j,J) = Z(edgepermute(J),nrElems+parents4e(j));
    coords = c4n_fine(nodes,:);
    area = area4e_fine(j);
    grads = [coords';1 1 1]\[1 0; 0 1; 0 0];
    Alocal = alpha4e(parents4e(j))*area*(grads*grads');
    dofs_loc = [nodes nrFineNodes+nodes];
    A_loc(1:3,1:3,j) = Alocal;
    A_loc(4:6,4:6,j) = Alocal;
    B_loc(:,:,j) = area*grads(:)*grads(:)';
    b1 = sigma4e(parents4e(j),1:2)*grads';
    b2 = sigma4e(parents4e(j),3:4)*grads';
    b(dofs_loc) = b(dofs_loc) + area*[b1 b2]';
end
DirichletNodes = unique(n4sDb_fine(:));
dofs = setdiff(1:nrNodes,DirichletNodes); nrDofs = length(dofs);
Iux4n(DirichletNodes,:) = u4Db(c4n_fine(DirichletNodes,:));
```

Listing A.15: Listing for Lines 1–42 of `estimateCREtaPMRED_Stokes.m`

A.7 Implementation of the PMRED Design

Listings A.15 and A.16 display the MATLAB code for the error estimator η_{PMRED} in its Stokes variant from Algorithm 5.2. The variant for the Poisson problem does not need to care about the divergence of the conforming approximation of u_{CR} and thus is easier to implement. Therefore, this Section presents the Stokes variant only. The input parameters include the inf-sup constant `c0` from (5.5) and the number of swings for the sum of norm minimisation, i.e., the number of iterations of the outer for loop in Algorithm 5.2. The default value is `nrSwings=3`.

Listing A.15 computes a piecewise linear function on the red-refined triangulation that equals the nodal interpolation of u_D along the Dirichlet boundary Γ_D and that equals u_{CR} in the midpoints of all other edges.

Listing A.15 computes the optimal values in the remaining vertices of the original triangulation as in Algorithm 5.2.

A.8 Implementation of the ARED Design

Listing A.17 displays the function `CR2RedP1` that maps a Crouzeix-Raviart function to a conforming piecewise affine function with respect to the red-refinement $\mathrm{red}(\mathcal{T})$. This corresponds to the design of v_{ARED} from Subsubsection 4.4.2.2 but without the correct Dirichlet boundary values. They have to be assigned outside the function.

The input vector `x` is of size $|\mathcal{E}| \times$ `k` and contains the values of the Crouzeix-Raviart function u_{CR} in the midpoints of the edges of the triangulation given through `c4n` and `n4e`. For Poisson problems as in Chapter 4, `k` equals `k=1` and for Stokes problems, `k` equals `k=2`.

The first output vector `x_fine(:,k)` contains the nodal values of the `k`-th component of the conforming averaging v_{ARED}. Line 25 assigns the averaged values for the node of the original triangulation and Line 26 copies the values of the Crouzeix-Raviart function stored in `x` to the values in the midpoints of the sides of the original triangulation which are the new nodes of the red-refinement. The averaging in (4.18) includes the weights $\lambda_{\max,T}$. They correspond to the input vector `alpha4e` of size $|\mathcal{T}| \times 1$.

The second output vector `x4e_fine` of size $|\mathrm{red}(\mathcal{T})| \times 3$ contains the elementwise nodal values of the Crouzeix-Raviart function u_{CR} (not v_{ARED}) with respect to $\mathrm{red}(\mathcal{T})$. This is needed for the computation of the broken energy norm $|||u_{\mathrm{CR}} - v_{\mathrm{ARED}}|||_{\mathrm{NC}}$. The entries of `x4e_fine` are assigned in Lines 15–23. The node numbers of the red-refinement in Line 19 that are larger than the number of nodes `nrNodes` of the original triangulation refer to the edge midpoints.

A.9 Implementation of the MP2 Nonconforming Error Estimator

The function `estimateCREtaMP2` computes the error estimator η_{MP2} for the optimal piecewise quadratic approximation of u_{CR} in the energy norm, see Subsubsection 4.4.2.3 for Poisson problems or Algorithm 5.1 for Stokes problems.

The function decides the case by the value of `c0`. If `c0=0` the function assumes a Poisson problem and otherwise a Stokes problem with `k=2` columns in the input vector `x` that should contain the coefficients of the Crouzeix-Raviart solution.

Lines 12–21 project the Crouzeix-Raviart function onto a piecewise quadratic function. The first three columns of `x4e_fine(:,1:3,k)` contain the nodal values of the `k`-th component of the Crouzeix-Raviart function, while the columns `x4e_fine(:,4:6,k)` contain its values in the edge midpoints. The next block consisting of Lines 22–52 computes the elementwise stiffness matrices of the piecewise quadratic polynomials as described in Subsubsection 2.3.5.4. Moreover, they compute the local contributions of the right-hand side vector

$$\texttt{b\_loc(m,j)} = \int_{T_m} \sigma_{\mathrm{CR}} \cdot \mathrm{D}\hat{\varphi}_j \quad \text{for } j = 1, \ldots, 6\texttt{k}.$$

```
e4n_fine = computeE4n(n4e_fine);
beta = 1;
for k=1:nrSwings
    for j=1:nrDofs
        node = dofs(j);
        elems = nonzeros(e4n_fine(:,node));
        globalnodes = unique(n4e_fine(elems,:));
        glob2locnodes = sparse(max(globalnodes));
        glob2locnodes(globalnodes) = 1:length(globalnodes);
        localdofs = [n4e_fine(elems,:) nrFineNodes+n4e_fine(elems,:)];
        dofT = localdofs';
        I = repmat(dofT(:),1,size(localdofs,2))';
        J = repmat(localdofs,1,size(dofT,1))';
        A = A_loc(:,:,elems); B = B_loc(:,:,elems);
        A = sparse(I(:),J(:),A(:),2*nrFineNodes,2*nrFineNodes);
        B = sparse(I(:),J(:),B(:),2*nrFineNodes,2*nrFineNodes);
        fixed_dofs = setdiff(globalnodes,node);
        fixed_dofs_loc = glob2locnodes(fixed_dofs);
        free_dofs_loc = [glob2locnodes(node) length(globalnodes)+glob2locnodes(node)];
        dofs_glob = [globalnodes' nrFineNodes+globalnodes'];
        A = A(dofs_glob,dofs_glob);
        B = B(dofs_glob,dofs_glob);
        b_loc = b(dofs_glob);
        x_loc = zeros(2*length(globalnodes),1);
        x_loc(fixed_dofs_loc) = Iux4n(fixed_dofs,1);
        x_loc(length(globalnodes)+fixed_dofs_loc) = Iux4n(fixed_dofs,2);
        M = (1+beta)*A + (1+1/beta)*B/c0^2;
        b_beta = (1+beta)*b_loc - M * x_loc;
        x_loc(free_dofs_loc) = M(free_dofs_loc,free_dofs_loc)\b_beta(free_dofs_loc);
        Iux4n(node,:) = x_loc(free_dofs_loc);
    end
    eta4e = zeros(size(n4e_fine,1),1);
    div4e = zeros(size(n4e_fine,1),1);
    for j = 1 : size(n4e_fine,1)
        nodes = n4e_fine(j,:);
        dc = [Iux4n(nodes,1); Iux4n(nodes,2)];
        div4e(j) = dc'*B_loc(:,:,j)*dc;
        x_loc = [x4e_fine1(j,:)' - Iux4n(nodes,1); x4e_fine2(j,:)' - Iux4n(nodes,2)];
        eta4e(j) = x_loc'*A_loc(:,:,j)*x_loc;
    end
    div = sqrt(sum(div4e))/c0;
    h1error = sqrt(sum(eta4e));
    beta = div/h1error;
end
for j = 1 : size(n4e_fine,1)
    nodes = n4e_fine(j,:);
    dc = [Iux4n(nodes,1); Iux4n(nodes,2)];
    div4e(j) = dc'*B_loc(:,:,j)*dc;
    x_loc = [x4e_fine1(j,:)' - Iux4n(nodes,1); x4e_fine2(j,:)' - Iux4n(nodes,2)];
    eta4e(j) = x_loc'*A_loc(:,:,j)*x_loc;
end
eta4e = accumarray(parents4e',eta4e);
div4e = accumarray(parents4e',div4e);
div = sqrt(sum(div4e));
eta = sqrt(sum(eta4e));
end
```

Listing A.16: Listing for Lines 43-end of `estimateCREtaPMRED_Stokes.m`

```
function [x_fine,x4e_fine] = CR2RedP1(c4n,n4e,n4sDb,n4sNb,x,alpha4e)
n4s = computeN4s(n4e);
s4e = computeS4e(n4e);
nrNodes = size(c4n,1);
nrSides = size(n4s,1);
nrComp = size(x,2);
[c4n_fine,n4e_fine,n4sDb_fine,n4sNb_fine,parents4e] = refineUniformRed(c4n,n4e,n4sDb,n4sNb);
weights4e = sqrt(alpha4e)*ones(1,3);
weightsum4n = accumarray(n4e(:),weights4e(:));
edgepermute = [2 3 1];
x_fine = zeros(nrNodes+nrSides,nrComp);
x4e_fine = zeros(size(n4e_fine,1),3,nrComp);
for k=1:nrComp
    xk = x(:,k);
    W = xk(s4e)';
    Z = (ones(3)-2*eye(3))*W;
    for j=1:size(n4e_fine,1)
        nodes = n4e_fine(j,:);
        [I,J] = find(nodes > nrNodes);
        x4e_fine(j,J,k) = xk(nodes(J)-nrNodes);
        [I,J] = find(nodes <= nrNodes);
        x4e_fine(j,J,k) = Z(edgepermute(J),parents4e(j));
    end
    Z = Z(edgepermute,:)'.*weights4e;
    x_fine(1:nrNodes,k) = accumarray(n4e(:),Z(:))./weightsum4n;
    x_fine(nrNodes+1:nrNodes+nrSides,k) = xk;
end
end
```

Listing A.17: Listing of `CR2RedP1.m`

There are six basis functions $\hat{\varphi}_j$ of $\mathcal{P}_2(T)$ for each component, so altogether twelve basis function of $\mathcal{P}_2(T;\mathbb{R}^2)$ in case of a Stokes problem. The discrete tensor σ_{CR} is represented by `sigma4e` and should equal the output array `sigma4e` of the corresponding solver. Since `sigma4e` is constant for each element it has the size $|\mathcal{T}| \times 2 \times$ `k`. So, the computation of `b_loc(m,1:6)` (or `b_loc(m,7:12)`) is a multiplication of `sigma4e(m,:,1)` (or `sigma4e(m,:,2)`) with the integrals over the gradients of the quadratic basis functions. Since the quadratic basis functions are combinations of the nodal basis functions $\varphi_1, \ldots, \varphi_3$ on a triangle, these integrals can be computed analytically with Lemma 2.2.18. For example, the integral of the gradient of the basis function $4\varphi_1\varphi_2$ equals

$$\int_T 4\nabla(\varphi_1\varphi_2)\,\mathrm{d}x = 4(\nabla\varphi_1 + \nabla\varphi_2)\,|T|/3$$

and the integral of the basis function of the form $\varphi_1(2\varphi_1 - 1)$ equals

$$\int_T \nabla(\varphi_1(2\varphi_1 - 1))\,\mathrm{d}x = \nabla\varphi_1\,|T|/3.$$

This led to the formulas in Lines 45–48. Lines 53–61 extend the local stiffness matrices to `k` components. Every additional component leads to a new block in the stiffness matrix that equals the block of the first component. The array `dofU4e` contains the numbers of the degrees of freedom for each element. For `k=1` component this equals `[n4e nrNodes+s4e]`. This also implies the enumeration of the global degrees of free-

dom, such that the first `nrNodes=size(n4e,1)` degrees of freedom refer to the nodes of the triangulation and the remaining `nrSides` degrees of freedom refer to the edges of the triangulation. For `k=2` components, there are additional `nrNodes+nrSides` degrees of freedom that are inserted behind the ones of the first component. Lines 62–66 accumulate all local contributions to the entries in the global stiffness matrix `A` and right-hand side vector `b`.

Part 2 in Listing A.19 commences with two if queries. The first one asks if the input parameter `nrCGsteps` is greater than zero. If this is the case, the user wants a pcg approximation $\eta_{\text{MP2cg(nrCGsteps)}}$ of η_{MP2}. Then, Line 69 computes the initial value with the function `CR2RedP1` from Listing A.17 which computes the nodal values for v_{ARED}. The next two lines fix the nodal values of the Dirichlet boundary nodes of the red-refinement. The next if query concerns the input parameter and inf-sup constant `c0`. If it is 0, a Poisson problem is assumed and the discrete problem `Ax=b` is solved exactly in Line 78 or with a pcg scheme in Line 83. The diagonal preconditioner is calculated in Line 80. Line 82 symmetrises the system matrix `A` which should be unnecessary in theory, but might be a good idea due to round-off errors and such.

If `c0>0`, a Stokes problem is assumed and we need an additional system matrix `B` that discretises the divergence bilinear form for the quadratic basis functions. This is done by numerical integration in Lines 86–88. The integrand in this integration is displayed in Lines 125–132 and employs the connection between the nodal basis functions and the quadratic basis functions. The for loop in Lines 90–110 realise Algorithm 5.1 for the minimisation of the sum of the broken energy norm and the L^2 norm of the divergence of $v_{\text{MP2cg(nrCGsteps)}}$. Finally, Lines 112–122 compute the piecewise and the total norms.

A.10 Implementation of the Overhead Terms

This section discusses the implementation of some overhead terms. Most of them can be found in the subfolder `integrate` and Table A.4 lists the filenames, what they compute and the minimal input data they require.

Filename	Computes what?	Input data
`DbError`	$\left\|h_{\mathcal{E}}^{3/2}\partial^2 u_D/\partial s^2\right\|_{L^2(\Gamma_D)}$	`TD2u4Db,c4n,n4e,n4sDb`
`DbErorrP2`	$\left\|h_{\mathcal{E}}^{3/2}\partial^2 (u_D - I_2 u_D)/\partial s^2\right\|_{L^2(\Gamma_D)}$	`TD2u4Db,c4n,n4e,n4sDb`
`hotLW`	$\|h_{\mathcal{T}}(f - f^\star)\|_{L^2(\Omega)}$	`f,c4n,n4e`
`hotLWNb`	$\left\|h_{\mathcal{E}}^{1/2}(g - g^\star)\right\|_{L^2(\Gamma_N)}$	`g,c4n,n4e,n4sNb`
`oscillations`	$\|h_{\mathcal{T}}(f - f_{\mathcal{T}})\|_{L^2(\Omega)}$	`f,c4n,n4e`
`oscillationsNb`	$\left\|h_{\mathcal{E}}^{1/2}(g - g_{\mathcal{E}})\right\|_{L^2(\Gamma_N)}$	`g,c4n,n4e,n4sNb`

Table A.4: List of functions in the `AFEM` package that compute overhead terms. The function handle `f` denotes the right-hand side source function, `g` denotes Neumann boundary data and `TD2u4Db` denotes the function handle for the second tangential derivative of the Dirichlet boundary data u_D.

```
function [eta4e,eta,div4e,div] = estimateCREtaMP2(u4Db,alpha4e,x,sigma4e,c4n,n4e,n4sDb,n4sNb,...
                                                  c0,nrSwings,nrCGsteps)
s4e     = computeS4e(n4e);
n4s     = computeN4s(n4e);
s4n     = computeS4n(n4e);
area4e  = computeArea4e(c4n,n4e);
Dbs     = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
mid4s   = computeMid4s(c4n,n4s);
nrElems = size(n4e,1); nrNodes = size(c4n,1);
nrSides = size(n4s,1); nrComp  = size(x,2);
x4e = zeros(nrElems,6,nrComp);
for k=1:nrComp
  xk = x(:,k);
  W = reshape(xk(s4e),[],size(s4e,2))';
  Z = ((ones(3)-2*eye(3))*W)';
  Z = Z(:,[2 3 1]);
  x4e(:,1:3,k) = Z(:,1:3);
  x4e(:,4,k) = sum(Z(:,1:2),2)/2;
  x4e(:,5,k) = sum(Z(:,2:3),2)/2;
  x4e(:,6,k) = sum(Z(:,[3 1]),2)/2;
end
b_loc = zeros(nrElems,6*nrComp);
A_loc = zeros(6,6,nrElems);
grads4e = zeros(3,2,nrElems);
for elem = 1 : nrElems
  nodes = n4e(elem,:);
  coords = c4n(nodes,:);
  area = area4e(elem);
  grads = [1 1 1;coords']\[0 0;eye(2)];
  AlocalP1 = alpha4e(elem) * area * grads * grads';
  AlocalP2(1:3,1:3) = AlocalP1.*(4*eye(3)-1);
  Atmp = 4*ones(3,1)*AlocalP1([6 3 2]);
  Atmp = Atmp-diag(diag(Atmp));
  AlocalP2(1:3,[5 6 4]) = Atmp;
  AlocalP2([5 6 4],1:3) = Atmp';
  d = diag(AlocalP1);
  d = d+d([2 3 1])+AlocalP1([2 6 3])';
  Atmp2 = diag(d);
  Atmp2([2 3 6]) = AlocalP1([3 6 2]);
  Atmp2 = Atmp2 + Atmp2' - diag(diag(Atmp2));
  AlocalP2(4:6,4:6) = 8*Atmp2;
  AlocalP2 = AlocalP2/3;
  A_loc(:,:,elem) = AlocalP2;
  for k=1:nrComp
    temp(1:3,k) = grads*sigma4e(elem,:,k)'*area/3;
    temp(4,k) = 4*(temp(1,k)+temp(2,k));
    temp(5,k) = 4*(temp(2,k)+temp(3,k));
    temp(6,k) = 4*(temp(3,k)+temp(1,k));
  end
  b_loc(elem,:) = temp(:);
  grads4e(:,:,elem) = grads;
end
m=nrNodes+nrSides; dofU4e = [n4e nrNodes+s4e];
c4n_fine = [c4n;mid4s]; fixedNodes = [unique(n4sDb(:)); nrNodes+Dbs];
val4fixedNodes = u4Db(c4n_fine(fixedNodes,:));
for j=2:nrComp;
  dofU4e = [dofU4e m*(j-1)+n4e m*(j-1)+nrNodes+s4e];
  A_loc((j-1)*6+(1:6),(j-1)*6+(1:6),:) = A_loc(1:6,1:6,:);
  fixedNodes = [fixedNodes; m*(j-1)+unique(n4sDb(:)); m*(j-1)+nrNodes+Dbs];
end
freeNodes = setdiff(1:nrComp*m,fixedNodes);
b = accumarray(dofU4e(:),b_loc(:));
dofU4eT = dofU4e';
I = repmat(dofU4eT(:),1,size(dofU4eT,1))';
J = repmat(dofU4e,1,size(dofU4eT,1))';
A = sparse(I(:),J(:),A_loc(:));
```

Listing A.18: Listing for Lines 1–66 of `estimateCRetaMP2.m`

```
xdiff4e = zeros(nrElems,6*nrComp);
if nrCGsteps > 0
  x0 = CR2RedP1(c4n,n4e,n4sDb,n4sNb,x,alpha4e);
  x0 = x0(:);
  x0(fixedNodes) = val4fixedNodes(:);
end
if c0 == 0
  y = zeros(size(A,1),1);
  y(fixedNodes) = val4fixedNodes(:);
  b = b - A*y;
  if nrCGsteps == 0
    y(freeNodes) = A(freeNodes,freeNodes)\b(freeNodes);
  else
    M1 = spdiags(1./spdiags(A(freeNodes,freeNodes),0),0,speye(size(A(freeNodes,freeNodes),1)));
    M2 = speye(length(freeNodes));
    M = (A + A')/2;
    [y(freeNodes),flag] = pcg(M(freeNodes,freeNodes),b(freeNodes),[],nrCGsteps,M1,M2,x0(freeNodes));
  end
else
  integrand = @(n4e,pts,pts_ref)P2DivergenceIntegrand(n4e,pts,pts_ref,grads4e);
  B_loc = permute(integrate(c4n,n4e,integrand,2),[3 2 1]);
  B = sparse(I(:),J(:),B_loc(:));
  beta = 1;
  for k=1:nrSwings
    y = zeros(nrComp*m,1);
    y(fixedNodes) = val4fixedNodes(:);
    M = (1+beta)*A + (1+1/beta)*B/c0^2;
    b_beta = (1+beta)*b - M * y;
    if nrCGsteps == 0
      y(freeNodes) = M(freeNodes,freeNodes)\b_beta(freeNodes);
    else
      M1 = spdiags(1./spdiags(M(freeNodes,freeNodes),0),0,speye(size(M(freeNodes,freeNodes),1)));
      M2 = speye(length(freeNodes));
      M = (M + M')/2;
      [y(freeNodes),flag] = pcg(M(freeNodes,freeNodes),b_beta(freeNodes),[],nrCGsteps,M1,M2,...
                                x0(freeNodes));
    end
    for n=1:nrComp
      xdiff4e(:,(1:6)+6*(n-1)) = x4e(:,:,n) - y(dofU4e(:,(1:6)+6*(n-1)));
    end
    h1error = sqrt(sum(matMul(matMul(permute(xdiff4e,[1 3 2]),permute(A_loc,[3 1 2])),xdiff4e)));
    div = sqrt(y'*B*y)/c0;
    beta = max(1e-12,div/h1error);
  end
end
y4e = zeros(nrElems,6*nrComp);
for k=1:nrComp
  y4e(:,(1:6)+6*(k-1)) = y(dofU4e(:,(1:6)+6*(k-1)));
  xdiff4e(:,(1:6)+6*(k-1)) = x4e(:,:,k) - y(dofU4e(:,(1:6)+6*(k-1)));
end
eta4e = matMul(matMul(permute(xdiff4e,[1 3 2]),permute(A_loc,[3 1 2])),xdiff4e);
eta = sqrt(sum(eta4e));
if nargout > 2
  div4e = matMul(matMul(permute(y4e,[1 3 2]),permute(B_loc,[3 1 2])),y4e);
  div = sqrt(sum(div4e));
end
end

function val = P2DivergenceIntegrand(n4e,pts,pts_ref,grads4e)
basisP1 = ones(size(n4e,1),1)*[1-sum(pts_ref) pts_ref(1) pts_ref(2)];
divergencesP1 = reshape(grads4e(:),[6 size(n4e,1)])';
divergencesP2(:,[1:3,7:9]) = divergencesP1.*(4*[basisP1 basisP1]-1);
divergencesP2(:,[4:6,10:12]) = 4*(basisP1(:,[1 2 3 1 2 3]).*divergencesP1(:,[2 3 1 5 6 4])...
                                + basisP1(:,[2 3 1 2 3 1]).*divergencesP1(:,[1 2 3 4 5 6]));
val = matMul(divergencesP2,permute(divergencesP2,[1 3 2]));
end
```

Listing A.19: Listing for Lines 67-end of `estimateCRetaMP2.m`

The function `hotLW` is discussed in Listing 3.2 of Subsection 3.2.2. Here we focus on the exemplary function `DbErorrP2`. This functions computes the Dirichlet boundary error for $\mathcal{P}_2(\mathcal{T})$-based interpolations of u_{CR} in Subsection 4.4.3 and I_2 denotes the nodal interpolation with respect to $\mathcal{P}_2(\mathcal{T})$. The main difference and difficulty in comparison with `DbError` is the computation of the second tangential derivative of $I_2 u_D$ that has to be subtracted from the second tangential derivative of u_D. To do so, Lines 12–16 of Listing A.20 compute the gradients and second derivatives of the three one-dimensional $\mathcal{P}_2$ basis functions on every Dirichlet edge $E \in \mathcal{E}(\Gamma_D)$. For an edge $E = (0, |E|)$, the first-order basis functions read $\varphi_1^{\mathcal{P}_1}(s) = 1 - s/|E|$ and $\varphi_2^{\mathcal{P}_1}(s) = s/|E|$. Their combination gives the second-order basis functions

$$\varphi_1^{\mathcal{P}_2}(s) = \varphi_1^{\mathcal{P}_1}(s)(2\varphi_1^{\mathcal{P}_1}(s) - 1), \quad \varphi_2^{\mathcal{P}_2}(s) = \varphi_2^{\mathcal{P}_1}(s)(2\varphi_2^{\mathcal{P}_1}(s) - 1) \quad \text{and}$$
$$\varphi_3^{\mathcal{P}_2}(s) = 4\varphi_1^{\mathcal{P}_1}(s)\varphi_2^{\mathcal{P}_1}(s).$$

Straightforward calculations yields the derivatives

$$\partial^2\varphi_1^{\mathcal{P}_2}/\partial s^2 = 4\left(\partial\varphi_1^{\mathcal{P}_1}/\partial s\right)^2, \quad \partial^2\varphi_2^{\mathcal{P}_2}/\partial s^2 = 4\left(\partial\varphi_2^{\mathcal{P}_1}/\partial s\right)^2 \quad \text{and}$$
$$\partial^2\varphi_3^{\mathcal{P}_2}/\partial s^2 = 8\left(\partial\varphi_1^{\mathcal{P}_1}/\partial s\right)\left(\partial\varphi_2^{\mathcal{P}_1}/\partial s\right).$$

The combination with the nodal values of u_D in the endpoints and the center of the edge in Line 20 yields the second tangential derivative of $I_2 u_D$, which is constant and stored in the array `constant4Dbs`. The remaining lines do the integration to calculate the edgewise contributions to

$$\left\| h_{\mathcal{E}}^{3/2}\partial^2(u_D - I_2u_D)/\partial s^2 \right\|_{L^2(\Gamma_D)} = \left(\sum_{E\in\mathcal{E}(\Gamma_D)} h_E^3 \left\|\partial^2(u_D - I_2u_D)/\partial s^2\right\|_{L^2(E)}^2 \right)^{1/2}.$$

The overhead terms for the obstacle problem are computed by `OBS_ExtraTerms` and `OBS_ExtraTermsNonAffine` (additional extra term for nonaffine obstacles) in the subfolder `estimate`.

The overhead term $\|\operatorname{div} u_h\|_{L^2(\Omega)}$ in the guaranteed upper bounds for the mini finite element method for the Stokes problem is computed by the solver `solveMINIStokes`.

A.11 Modifications for Curved Boundaries

The numerical experiments with curved boundaries cause several modifications to the data structures, the error estimators and the refinement procedures. These modifications were collected together in a subfolder with the name `curved_boundaries`.

The main modifications concern a new data field `n4sCb`, similar to `n4sDb` and `n4sNb`, that contains pairs of node numbers of sides that relate to the curved boundary. Whenever such a side in `n4sCb` is bisected by a refinement routine in `curved_boundaries`, the new side midpoint is shifted onto the unit sphere and the two subsides are added into the `n4sCb` field for the refined triangulation.

```
function [etaDb,etaDb4s,e4Dbs] = DbErrorP2(TD2u4Db,u4Db,c4n,n4e,n4sDb,degree,nrComp)
s4n         = computeS4n(n4e);
Dbs         = rowaddr(s4n,n4sDb(:,1),n4sDb(:,2));
e4s         = computeE4s(n4e);
e4Dbs       = e4s(Dbs,1);
n4s         = computeN4s(n4e);
length4Dbs  = computeLength4s(c4n,n4sDb);
mid4Dbs     = computeMid4s(c4n,n4sDb);
tangent4Dbs = computeTangent4s(c4n,n4s(Dbs,:));
constant4Dbs= zeros(size(n4sDb,1),nrComp);
for j=1:size(n4sDb,1)
    nodes = n4sDb(j,:); length = length4Dbs(j);
    grads(1) = -1/length; grads(2) = 1/length;
    grads4Dbs(1) = 4*grads(1)*grads(1);
    grads4Dbs(2) = 4*grads(2)*grads(2);
    grads4Dbs(3) = 8*grads(1)*grads(2);
    u4Db4n = u4Db(c4n(nodes,:)); u4Db4s = u4Db(mid4Dbs(j,:));
    for k=1:nrComp
        vals([1 2]) = u4Db4n(:,k); vals(3) = u4Db4s(:,k);
        constant4Dbs(j,k) = sum(vals.*grads4Dbs);
    end
end
integrand = @(n4p,pts,pts_ref)sum((TD2u4Db(pts,tangent4Dbs)-constant4Dbs).^2,2);
etaDb4s = length4Dbs.^3.*integrate(c4n,n4sDb,integrand,degree);
etaDb = sqrt(sum(etaDb4s));
end
```

Listing A.20: Listing of `DbErrorP2.m`

The further modifications relate to exact integration on circle segments. For this reason, the new quadrature routine `integratePolar` of Listing A.21 was written. It allows to integrate over a domain that can be parameterised by

$$\omega := \{(x,y) = (x_0,y_0) + r(\cos\varphi, \sin\varphi) \in \mathbb{R}^2 \,|\, \alpha \leqslant \varphi \leqslant \beta, R_{\text{low}}(\varphi) \leqslant r \leqslant R_{\text{up}}(\varphi)\}.$$

Here the point (x_0, y_0) is the center for the polar coordinate parameterisation and R_{low} and R_{up} are functions that depend on φ and denote the minimal and maximal radius. The integrand v under the integrals

$$\int_\alpha^\beta \int_{R_{\text{low}}(\varphi)}^{R_{\text{low}}(\varphi)} v(r\cos\varphi, r\sin\varphi) r \,\mathrm{d}r \,\mathrm{d}\varphi \tag{A.1}$$

is evaluated in Cartesian coordinates $(x_1, x_2) = r(\cos(\varphi), \sin(\varphi))$.

Each integral in (A.1) is approximated by 1D Gauss-Legendre quadrature. The subfunction `getGaussPoints` in Lines 25–32 compute the `N` Gauss-Legendre points and their weights on the unit interval. In Lines 3–4 they are transformed to the vector `phi` for the interval `[alpha,beta]`. Then, a loop over the `N` entries in `phi` is in order. For every arc `phi(j)`, the interval for the radius is determined by evaluation of the two function handles `Rphi_low` and `Rphi_up` in Lines 7 and 8. The next two lines transform the unit Gauss points to the Gauss-Legendre points for the radius interval `[Rphi_low(phi(j)),Rphi_up(phi(j))]`. To evaluate the function handle for the integrand, the polar coordinates of the Gauss points are transformed into Cartesian coordinates. Here we have to add the center of the polar coordinates. If the optional

```
function val = integratePolar(alpha,beta,Rphi_low,Rphi_up,v,c4n_center,N,c4n_elem)
[x0,w0] = getGaussPoints(N);
phi = x0*(beta-alpha) + alpha;
wphi = w0*(beta-alpha);
val = 0;
for j=1:N
    r_up = Rphi_up(phi(j));
    r_low = Rphi_low(phi(j));
    r = x0*(r_up-r_low) + r_low;
    wr = w0*(r_up-r_low);
    [x,y] = pol2cart(phi(j),r);
    x = x + c4n_center(1);
    y = y + c4n_center(2);
    if ~isempty(c4n_elem)
        A = [c4n_elem(:,1)';c4n_elem(:,2)';[1 1 1]];
        x_ref = A\[x';y';ones(1,length(x))];
        x_ref = x_ref([1 2],:)';
    else
        x_ref = [0 0];
    end
    val = val + wphi(j)*sum(wr.*r.*v([x y]));
end
end

function [x,w] = getGaussPoints(n)
    gamma = (1 : n-1) ./ sqrt(4*(1 : n-1).^2 - ones(1,n-1) );
    [V,D] = eig( diag(gamma,1) + diag(gamma,-1) );
    x = diag(D);
    w = 2*V(1,:).^2;
    x = .5 * x + .5;
    w = .5 * w';
end
```

Listing A.21: Listing of `integratePolar.m`

```
n4e = [1 2 3];
c4n = [0.92387953251129   0.38268343236509
       0.38268343236509   0.92387953251129
       0.25               0.25];
f = @(x)(1);
plotTriangulation(c4n,n4e); hold on;
phi = 0:0.01:2*pi;
plot(cos(phi),sin(phi),'Color',[0.5 0.5 0.5]); axis square;
R_up = @(ph)Rphi_up(ph,c4n(3,:));
phi = cart2pol(c4n([1 2],1)-c4n(3,1),c4n([1 2],2)-c4n(3,2));
J = find(phi < 0);
phi(J) = 2*pi + phi(J);
integratePolar(phi(1),phi(2),@(x)(0),R_up,@(x,x_ref)f(x),c4n(n4e(1,:),:),c4n(3,:),16)

function val = Rphi_up(phi,a)
  w = a(1)*cos(phi)+a(2)*sin(phi);
  q = a(1)^2 + a(2)^2 - 1;
  val = -w + sqrt(w^2-q);
end
```

Listing A.22: Example for usage of `integratePolar`

input parameter `c4n_elem` is assigned, also the reference coordinates with respect to this element are calculated. Note that they do not have to assume values in $[0,1]$ if the domain specified by the given arc and radius intervals is larger. Line 21 evaluates the integrand in all Gauss points of the complete radius interval in parallel and multiplies the result with the weights for the Gauss points. The sum over all Gauss points yields the approximation of the integral.

N	Quadrature error
1	-1.570796326794897
2	0.503950238354575
4	0.002236469020655
8	0.000000000182882
16	-0.000000000000000
32	0.000000000000004
64	0.000000000000001
128	0.000000000000002
256	0.000000000000000

Table A.5: Error for the approximation of the integral $\int_{B(0,1)} x_1 \,\mathrm{d}x_1 \,\mathrm{d}x_2 = 0$ as computed by `integratePolar(0,2*pi,@(x)0,@(x)1,@(x,x_ref)x(:,1),[0 0],N)` for different values of `N`.

To test the accuracy of `integratePolar`, we integrate the function $f(x) = x_1$ (given as the function handle `f = @(x,x_ref)(x(:,1))`) over the unit sphere $B(0,1)$ with radius 1. This is done with the `MATLAB` code line

```
integratePolar(0,2*pi,@(x)0,@(x)1,f,[0 0],N);
```

The parameter `N` denotes the number of Gauss points. Easy analytic calculations show

$$\int_{B(0,1)} x_1 \,\mathrm{d}x_1 \,\mathrm{d}x_2 = \int_0^{2\pi} \int_0^1 r\cos(\varphi) r \,\mathrm{d}r \,\mathrm{d}\varphi = 0.$$

Table A.5 shows the error for different values of `N`. It conveys that there is a fast convergence towards the exact value within the limits of the computational accuracy.

To give a more involved example, consider a triangle $T = \mathrm{conv}(P_1, P_2, P_3)$ with P_1 and P_2 on the circle with radius 1 and center $(0,0)$ and P_3 somewhere inside this circle. Furthermore, consider the circle segment S that is cut off from the circle by the line between P_1 and P_2. The code in Listing A.22 plots such a triangle and the circle and integrates the function $f \equiv 1$ over $T \cup S$. To do so, the start and end arcs for `integratePolar` are set to `alpha = phi(1)` and `beta = phi(2)` where `phi` contains the arc of the polar coordinates of $P_1 - P_3$ and $P_2 - P_3$, respectively. This implies that `c4n_center=c4n(3,:)` $= P_3$ is the center of the polar coordinates. The radius assumes values between 0 (hence `Rphi_low = @(x)(0)`) and the maximal radius, which is computed by the function handle `R_up` in Lines 15–19 by the evaluation of some analytic formula. These are the input data for `integratePolar` in Line 13 together with the number of Gauss points `N=16`.

B Common Notation

The following table lists notation that appear regularly in the thesis. If adequate also the page number of their first occurrence is given.

Elementary Notation

$\mathbb{I}$		the identity or unit element with respect to multiplication/concatenation for matrices/function operators
$A \lesssim B$		$A \leqslant CB$ for some multiplicative constant C that depends on the domain Ω or on the shape, but not on the mesh size of the finite elements
$A \approx B$		$A \lesssim B$ and $B \lesssim A$
$[a, b]$		*closed* interval $\{x \in \mathbb{R} \,\big|\, a \leqslant x \leqslant b\}$
$a \cdot b$		ℓ^2 scalar product $\sum_{j=1}^{n} a_j b_j$ of two vectors $a, b \in \mathbb{R}^n$
$A : B$		matrix inner product $\sum_{j=1}^{n} \sum_{k=1}^{m} A_{jk} B_{jk}$ for two matrices $A, B \in \mathbb{R}^{n \times m}$.
$\subseteq$		subset; equality is permitted
$\lvert X \rvert$		absolute value of a real number X; ℓ^2 norm of a vector X; Frobenius norm $(X : X)^{1/2}$ of a matrix X; positive Lebesgue-measure of a set X
conv		convex hull
$\operatorname{diam}(\omega)$		diameter of a domain ω
$\operatorname{width}(\omega)$		width of a domain ω
$\operatorname{eig}(A)$		set of eigenvalues or spectrum of a symmetric postive definite matrix A
$\operatorname{tr}(A)$		trace of a matrix A
A^{T}		transpose of a matrix A
$\operatorname{dist}(x, E)$		distance $\operatorname{dist}(x, E) = \min_{y \in E} \lvert x - y \rvert$ between x and E
Δv		Laplace operator, $\Delta v = \sum_{j=1}^{n} \partial^2 v / \partial x_j^2$
∇		gradient operator for scalar-valued (weak) differentiable functions
D		gradient operator for vector-valued (weak) differentiable functions
$\operatorname{div} v$		divergence operator $\sum_{j=1}^{n} \partial v_j / \partial x_j$ for functions $v \in H(\operatorname{div}, \Omega)$
$\operatorname{Curl} v$	p. 14	curl operator for functions $v \in H^1(\Omega; \mathbb{R}^s)$ where s depends on the dimension of Ω, see (2.1)
dev	p. 103	deviatoric part of a tensor

v_ω	integral mean $\fint_\omega v \,\mathrm{d}x$ of v over a domain ω
$v_\mathcal{T}$	piecewise integral mean of v $v_\mathcal{T}\|_T = v_T$ for all $T \in \mathcal{T}$ for a triangulation $\mathcal{T}$
$v_\mathcal{E}$	piecewise integral mean of v $v_\mathcal{E}\|_E = v_E$ for all $E \in \mathcal{E}$ for a set of sides $\mathcal{E}$
ν, ν_ω	normal vector, outer normal vector of the domain ω
ν_E	oriented normal vector of a side E
τ	tangent vector

Norms and Function Spaces

$\|v\|_{L^2(\omega)}$		L^2 norm $\int_\omega \|v\|^2 \,\mathrm{d}x$ on ω
$\|\|\|v\|\|\|$	p. 24	energy norm $\int_\Omega \mathbb{S}\nabla v \cdot \nabla v \,\mathrm{d}x$
$\|\|\|F\|\|\|_{W^\star}$	p. 37	dual norm $\sup_{v\in W} F(v)/\|\|\|v\|\|\|$ of a functional $F \in W^\star$ with respect to a space $W \subseteq H^1(\Omega)$
$C(\omega)$		continuous functions on ω
$C_D(\Omega)$		continuous functions on Ω with zero values along Γ_D
$L^2(\omega)$		square-integrable functions on ω with $\|v\|_{L^2(\omega)} < \infty$
$H^1(\omega)$	p. 11	square-integrable functions on ω with weak gradient in L^2
$H^1_D(\Omega)$		set of functions $v \in H^1(\Omega)$ with $v = 0$ along the Dirichlet boundary Γ_D
$H^1(\Omega)/\mathbb{R}$		set of functions $v \in H^1(\Omega)$ with $\int_\Omega v \,\mathrm{d}x = 0$
V		set of test functions, usually $V = H^1_D(\Omega)$ if $\|\Gamma_D\| > 0$ or $V = H^1(\Omega)/\mathbb{R}$ if $\Gamma_D = \varnothing$
$H(\mathrm{div}, \omega)$	p. 11	square-integrable functions on ω with divergence in L^2
$\mathcal{K}$	p. 137	set of admissable functions in the obstacle problem
$\mathcal{P}_k(\mathcal{T}), \mathcal{P}_k(\mathcal{E})$	p. 15	piecewise polynomials of order k with respect to a triangulation $\mathcal{T}$ or set of sides $\mathcal{E}$
$V(\mathcal{T})$	p. 24	set of $\mathcal{P}_1$-conforming functions, $V(\mathcal{T}) = \mathcal{P}_1(\mathcal{T}) \cap C_D(\Omega)$, with respect to a triangulation $\mathcal{T}$
$\mathrm{CR}(\mathcal{T})$	p. 20	set of Crouzeix-Raviart functions of lowest order with respect to a triangulation $\mathcal{T}$
$\mathrm{CR}_0(\mathcal{T})$	p. 24	set of Crouzeix-Raviart functions of lowest order that are zero in the midpoints of the Dirichlet sides $E(\Gamma_D)$ of the triangulation $\mathcal{T}$
$\mathrm{RT}_0(\mathcal{T})$	p. 20	set of Raviart-Thomas functions of lowest order with respect to a triangulation $\mathcal{T}$
$Q(\mathcal{T})$	p. 26	set of Raviart-Thomas functions of lowest order that have zero normal fluxes along the Neumann sides $E(\Gamma_N)$ of the triangulation $\mathcal{T}$
$\mathrm{Mini}(\mathcal{T})$	p. 105	set of discrete functions for the mini finite element method

$\mathcal{K}(\mathcal{T})$	p. 137	discrete set of admissable functions of the finite element method for the obstacle problem
φ_z	p. 19	nodal basis function of a node $z \in \mathcal{N}$
ψ_E	p. 19	Crouzeix-Raviart basis function of a side $E \in \mathcal{E}$
ϑ_E	p. 19	Raviart-Thomas basis function of some side $E \in \mathcal{E}$

Triangulations

$\mathcal{T}$	p. 17	regular triangulation of a domain Ω
$\mathcal{T}^\star$	p. 41	dual triangulation of $\mathcal{T}$
$\mathrm{red}(\mathcal{T})$	p. 29	red-refinement of a regular triangulation $\mathcal{T}$
$\mathcal{N}$	p. 17	set of nodes of a regular triangulation $\mathcal{T}$
$\mathcal{E}$	p. 17	set of sides (edges in 2D or faces in 3D) of a regular triangulation $\mathcal{T}$
$\mathcal{N}(\Gamma_D)$	p. 18	set of Dirichlet boundary nodes
$\mathcal{E}(\Gamma_D)$	p. 18	set of Dirichlet boundary sides
$\mathcal{M}$	p. 18	set of free nodes of a regular triangulation $\mathcal{T}$, $\mathcal{M} = \mathcal{N} \backslash \mathcal{N}(\Gamma_D)$
$\omega_z, \omega_E, \omega_T$	p. 18	neighbourhood patches of a node $z \in \mathcal{N}$, a side $E \in \mathcal{E}$ or a element $T \in \mathcal{T}$
h_z		diameter $\mathrm{diam}(\omega_z)$ of the node patch for $z \in \mathcal{N}$
h_T		diameter $\mathrm{diam}(T)$ of a element $T \in \mathcal{T}$
$h_\mathcal{T}$		local mesh size or piecewise diameter, $h_\mathcal{T}\|T = h_T$ for all $T \in \mathcal{T}$

Constants

$C_\mathrm{P}(\omega)$	p. 13	Poincaré constant on a domain ω from Theorem 2.1.8
$C_\mathrm{F}(\omega)$	p. 13	Friedrichs constant on a domain ω from Theorem 2.1.10
$j_{1,1}$	p. 13	first positive root of the first Bessel function, $j_{1,1} = 3.8317\ldots$
$C_\mathrm{N}(E)$	p. 44	constant for a Neumann boundary side $E \in \mathcal{E}$ from Theorem 3.2.2
$C_{\mathrm{D},1}(E), C_{\mathrm{D},2}(E)$	p. 67	constants for a Dirichlet boundary side $E \in \mathcal{E}$ from Theorem 4.2.2
hot		higher-order terms

Bibliography

M. Ainsworth. Robust a posteriori error estimation for nonconforming finite element approximation. *SIAM J. Numer. Anal.*, 42(6):2320–2341, 2004. ISSN 0036-1429. doi: http://dx.doi.org/10.1137/S0036142903425112.

M. Ainsworth. A posteriori error estimation for lowest order Raviart-Thomas mixed finite elements. *SIAM J. Sci. Comput.*, 30(1):189–204, 2007/08. ISSN 1064-8275.

M. Ainsworth and W. Dörfler. Reliable a posteriori error control for nonconformal finite element approximation of Stokes flow. *Math. Comp.*, 74(252):1599–1619 (electronic), 2005. ISSN 0025-5718.

M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000. ISBN 0-471-29411-X. doi: 10.1002/9781118032824.

D. N. Arnold and R. S. Falk. A uniformly accurate finite element method for the Reissner-Mindlin plate. *SIAM J. Numer. Anal.*, 26(6):1276–1290, 1989. ISSN 0036-1429. doi: 10.1137/0726074.

I. Babuška and T. Strouboulis. *The finite element method and its reliability*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 2001.

C. Bahriawati and C. Carstensen. Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control. *Comput. Methods Appl. Math.*, 5 (4):333–361 (electronic), 2005. ISSN 1609-4840.

R. E. Bank and B. D. Welfert. A posteriori error estimates for the Stokes problem. *SIAM J. Numer. Anal.*, 28(3):591–623, 1991. ISSN 0036-1429.

S. Bartels and C. Carstensen. Averaging techniques yield reliable a posteriori finite element error control for obstacle problems. *Numer. Math.*, 99(2):225–249, 2004. ISSN 0029-599X. doi: 10.1007/s00211-004-0553-6.

S. Bartels, C. Carstensen, and G. Dolzmann. Inhomogeneous Dirichlet conditions in a priori and a posteriori finite element error analysis. *Numer. Math.*, 99(1):1–24, 2004. ISSN 0029-599X.

M. Bebendorf. A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen*, 22(4):751–756, 2003. ISSN 0232-2064.

C. Bernardi and R. Verfürth. Adaptive finite element methods for elliptic equations with non-smooth coefficients. *Numer. Math.*, 85(4):579–608, 2000. ISSN 0029-599X. doi: 10.1007/PL00005393.

D. Braess. A posteriori error estimators for obstacle problems—another look. *Numer. Math.*, 101(3):415–421, 2005. ISSN 0029-599X. doi: 10.1007/s00211-005-0634-1.

D. Braess. *Finite elements*. Cambridge University Press, Cambridge, third edition, 2007. ISBN 978-0-521-70518-9. doi: 10.1017/CBO9780511618635. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.

D. Braess. An a posteriori error estimate and a comparison theorem for the nonconforming P_1 element. *Calcolo*, 46(2):149–155, 2009. ISSN 0008-0624.

D. Braess and J. Schöberl. Equilibrated residual error estimator for edge elements. *Math. Comp.*, 77(262):651–672, 2008. ISSN 0025-5718.

J. Brandts, Y. Chen, and J. Yang. A note on least-squares mixed finite elements in relation to standard and mixed finite elements. *IMA J. Numer. Anal.*, 26(4):779–789, 2006. ISSN 0272-4979. doi: 10.1093/imanum/dri048.

S. C. Brenner and C. Carstensen. *Finite Element Methods*. John Wiley and Sons, 2004.

S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008. ISBN 978-0-387-75933-3. doi: 10.1007/978-0-387-75934-0.

F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991. ISBN 0-387-97582-9. doi: 10.1007/978-1-4612-3172-1.

C. Carstensen. Quasi-interpolation and a posteriori error analysis in finite element methods. *M2AN Math. Model. Numer. Anal.*, 33(6):1187–1202, 1999. ISSN 0764-583X. doi: 10.1051/m2an:1999140.

C. Carstensen. An adaptive mesh-refining algorithm allowing for an H^1 stable L^2 projection onto Courant finite element spaces. *Constr. Approx.*, 20(4):549–564, 2004. ISSN 0176-4276.

C. Carstensen. A unifying theory of a posteriori finite element error control. *Numer. Math.*, 100(4):617–637, 2005. ISSN 0029-599X. doi: http://dx.doi.org/10.1007/s00211-004-0577-y.

C. Carstensen. Finite element method. Yonsei Lectures at the WCU Department Computational Science and Engineering. Yonsei University, 120-749 Seoul, Korea. Unpublished Lecture Notes., 2009.

C. Carstensen and S. Bartels. Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM. *Math. Comp.*, 71(239):945–969 (electronic), 2002. ISSN 0025-5718. doi: 10.1090/S0025-5718-02-01402-3.

C. Carstensen and S. A. Funken. Fully reliable localised error control in the fem. *SIAM J. Sci. Comput.*, 21(4):1465–1484 (electronic), 1999. ISSN 1095-7197.

C. Carstensen and S. A. Funken. A posteriori error control in low-order finite element discretisations of incompressible stationary flow problems. *Math. Comp.*, 70(236):1353–1381 (electronic), 2001. ISSN 0025-5718.

C. Carstensen and J. Gedicke. Guaranteed lower bounds for eigenvalues. 2013+. Submitted.

C. Carstensen and C. Merdon. Estimator competition for Poisson problems. *J. Comp. Math.*, 28(3):309–330 (electronic), 2010.

C. Carstensen and C. Merdon. A posteriori error estimator competition for conforming obstacle problems. *Numer. Methods Partial Differential Equations*, 2012. doi: 10.1002/num.21728.

C. Carstensen and C. Merdon. Computational survey on a posteriori error estimators for nonconforming finite element methods for Poisson problems. *J. Comput. Appl. Math*, 2013. doi: 10.1016/j.cam.2012.12.021.

C. Carstensen and C. Merdon. Refined fully explicit a posteriori residual-based error control. 2013+. Submitted.

C. Carstensen and C. Merdon. Effective postprocessing for equilibration a posteriori error estimators. *Numer. Math.*, 123(3):425–459, 2013.

C. Carstensen and Numerical Analysis Group, HU Berlin. AFEM. unpublished MATLAB software package, 2009.

C. Carstensen, M. Eigel, R. H. W. Hoppe, and C. Loebhard. Numerical mathematics: Theory, methods and applications. *Numer. Math. Theor. Meth. Appl.*, 5(4):509–558, 2012a. doi: 10.4208/nmtma.2011.m1032.

C. Carstensen, D. Peterseim, and M. Schedensack. Comparison results of three first-order finite element methods for the poisson model problem. 2012b.

P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam, 1978. ISBN 0-444-85028-7. Studies in Mathematics and its Applications, Vol. 4.

E. Dari, R. Durán, and C. Padra. Error estimators for nonconforming finite element approximations of the Stokes problem. *Math. Comp.*, 64(211):1017–1033, 1995. ISSN 0025-5718. doi: 10.2307/2153481.

E. Dari, R. Durán, C. Padra, and V. Vampa. A posteriori error estimators for nonconforming finite element methods. *RAIRO Modél. Math. Anal. Numér.*, 30(4):385–400, 1996. ISSN 0764-583X.

L. C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010. ISBN 978-0-8218-4974-3.

L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992. ISBN 0-8493-7157-0.

R. S. Falk. Error estimates for the approximation of a class of variational inequalities. *Math. Comp.*, 28:963–971, 1974. ISSN 0378-4754.

S. A. Funken. *Beiträge zur a posteriori Fehlerabschätzung bei der numerischen Behandlung elliptischer partieller Differentialgleichungen - Theorie, Numerik und Anwendungen*. Habilitation, Universität Kiel, 2002.

G. P. Galdi. *An introduction to the mathematical theory of the Navier-Stokes equations. Vol. II*, volume 39 of *Springer Tracts in Natural Philosophy*. Springer-Verlag, New York, 1994. ISBN 0-387-94150-9. doi: 10.1007/978-1-4612-5364-8. Nonlinear steady problems.

V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986. ISBN 3-540-15796-4. doi: 10.1007/978-3-642-61623-5. Theory and algorithms.

C. Gräser and R. Kornhuber. Multigrid methods for obstacle problems. *J. Comput. Math.*, 27(1):1–44, 2009. ISSN 0254-9409.

P. Grisvard. *Elliptic problems in nonsmooth domains*, volume 24 of *Monographs and Studies in Mathematics*. Pitman (Advanced Publishing Program), Boston, MA, 1985. ISBN 0-273-08647-2.

W. Han. *A posteriori error analysis via duality theory*, volume 8 of *Advances in Mechanics and Mathematics*. Springer-Verlag, New York, 2005. ISBN 0-387-23536-1.

D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications*, volume 88 of *Pure and Applied Mathematics*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1980. ISBN 0-12-407350-6.

R. S. Laugesen and B. A. Siudeja. Minimizing Neumann fundamental tones of triangles: an optimal Poincaré inequality. *J. Differential Equations*, 249(1):118–135, 2010. ISSN 0022-0396. doi: 10.1016/j.jde.2010.02.020.

R. Luce and B. I. Wohlmuth. A local a posteriori error estimator based on equilibrated fluxes. *SIAM J. Numer. Anal.*, 42(4):1394–1414 (electronic), 2004. ISSN 0036-1429. doi: 10.1137/S0036142903433790.

S. Mao, X. Zhao, and Z. Shi. Convergence of a standard adaptive nonconforming finite element method with optimal complexity. *Appl. Numer. Math.*, 60(7):673–688, 2010. ISSN 0168-9274. doi: 10.1016/j.apnum.2010.03.010.

R. H. Nochetto and L. B. Wahlbin. Positivity preserving finite element approximation. *Math. Comp.*, 71(240):1405–1419 (electronic), 2002. ISSN 0025-5718. doi: 10.1090/S0025-5718-01-01369-2.

R. H. Nochetto, K. G. Siebert, and A. Veeser. Pointwise a posteriori error control for elliptic obstacle problems. *Numer. Math.*, 95(1):163–195, 2003. ISSN 0029-599X. doi: 10.1007/s00211-002-0411-3.

L. E. Payne and H. F. Weinberger. An optimal Poincaré inequality for convex domains. *Arch. Rat. Mech. Anal.*, 5:286–292, 1960.

W. Prager and J. L. Synge. Approximations in elasticity based on the concept of function space. *Quart. Appl. Math.*, 5:241–269, 1947. ISSN 0033-569X.

S. Repin. *A Posteriori Estimates for Partial Differential Equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter, Berlin, 2008.

S. I. Repin. A unified approach to a posteriori error estimation based on duality error majorants. *Math. Comput. Simulation*, 50(1-4):305–321, 1999. ISSN 0378-4754. doi: 10.1016/S0378-4754(99)00081-6. Modelling '98 (Prague).

G. Stoyan. Towards discrete Velte decompositions and narrow bounds for inf-sup constants. *Comput. Math. Appl.*, 38(7-8):243–261, 1999. ISSN 0898-1221.

A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1971. Prentice-Hall Series in Automatic Computation.

R. Temam. *Navier-Stokes equations*. AMS Chelsea Publishing, Providence, RI, 2001. ISBN 0-8218-2737-5. Theory and numerical analysis, Reprint of the 1984 edition.

J. Valdman. Minimization of functional majorant in a posteriori error analysis based on H(div) multigrid-preconditioned cg method. *Advances in Numerical Analysis*, 2009, 2009.

A. Veeser. Efficient and reliable a posteriori error estimators for elliptic obstacle problems. *SIAM J. Numer. Anal.*, 39(1):146–167 (electronic), 2001. ISSN 0036-1429.

A. Veeser and R. Verfürth. Explicit upper bounds for dual norms of residuals. *SIAM J. Numer. Anal.*, 47(3):2387–2405, 2009. ISSN 0036-1429.

A. Veeser and R. Verfürth. Poincaré constants for finite element stars. *IMA J. Numer. Anal.*, 32(1):30–47, 2012. ISSN 0272-4979. doi: 10.1093/imanum/drr011. URL `http://dx.doi.org/10.1093/imanum/drr011`.

R. Verfürth. A posteriori error estimators for the Stokes equations. *Numer. Math.*, 55(3): 309–325, 1989. ISSN 0029-599X.

R. Verfürth. A posteriori error estimates for nonlinear problems. Finite element discretizations of elliptic equations. *Math. Comp.*, 62(206):445–475, 1994. ISSN 0025-5718. doi: 10.2307/2153518.

R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*. Wiley-Teubner, 1996.

M. Vohralík. A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 45(4):1570–1599 (electronic), 2007. ISSN 0036-1429.

M. Vohralík. Guaranteed and fully robust a posteriori error estimates for conforming discretizations of diffusion problems with discontinuous coefficients. *J. Sci. Comput.*, 46 (3):397–438, 2011. ISSN 0885-7474.

List of Figures

List of Tables

List of Algorithms

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Berlin, den 28. Feb. 2013 Christian Merdon